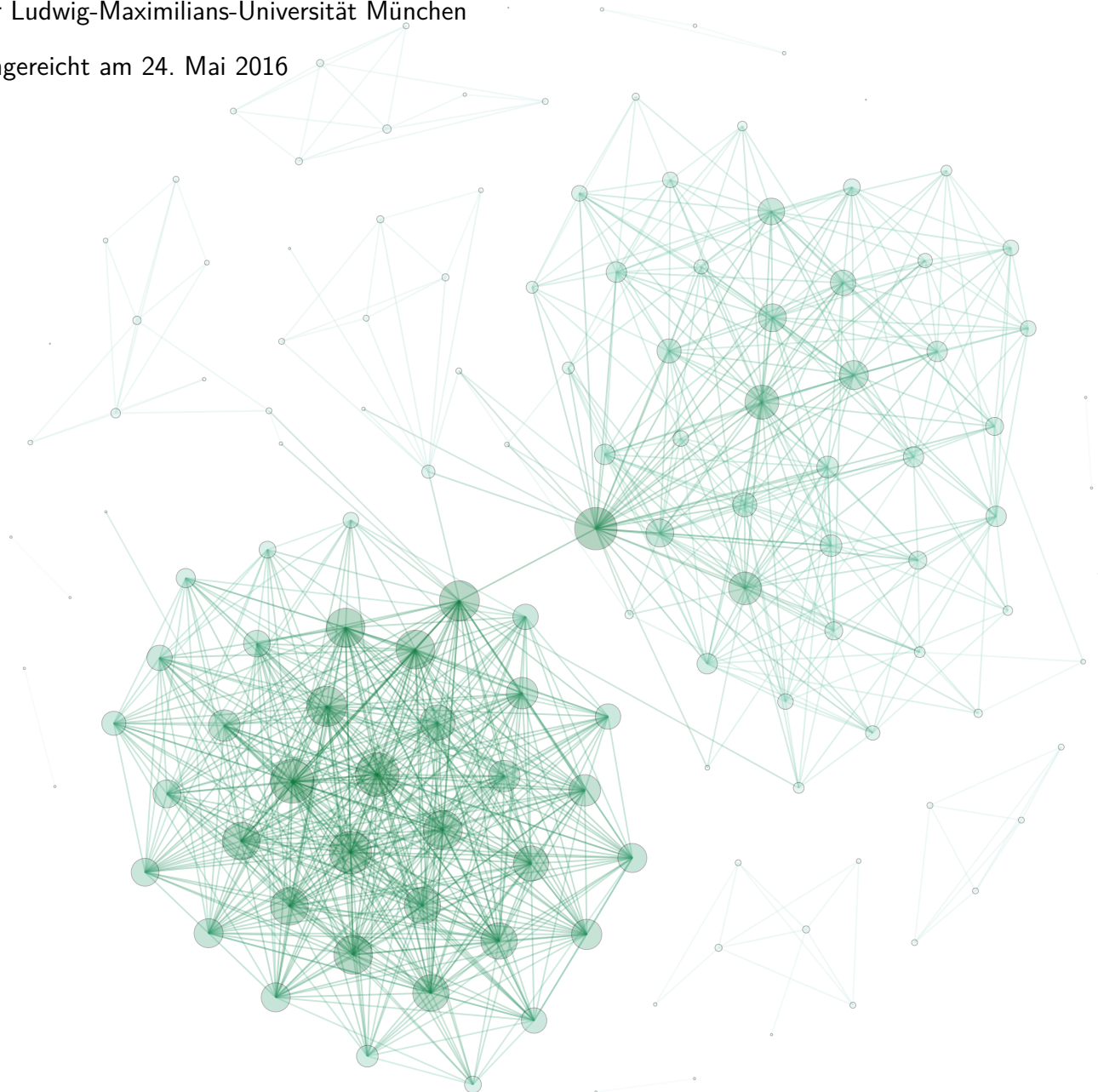


Stephanie Thiemichen

Extensions of Exponential Random Graph Models for Network Data Analysis

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 24. Mai 2016



Stephanie Thiemichen

Extensions of Exponential Random Graph Models for Network Data Analysis

Dissertation an der Fakultät für Mathematik, Informatik und Statistik
der Ludwig-Maximilians-Universität München

Eingereicht am 24. Mai 2016

Erster Berichterstatter: Prof. Dr. Göran Kauermann
Zweiter Berichterstatter: Prof. Dr. Ernst Wit

Tag der Disputation: 15. Juli 2016

für Annelies und Kurt

Danksagung

Mein besonderer Dank gilt allen, die zur Entstehung dieser Doktorarbeit in der ein oder anderen Form beigetragen haben. Das sind ...

...mein Doktorvater Göran Kauermann. Vielen Dank für die Betreuung und das gemeinsame Durchstehen aller Probleme, die sich beim Einarbeiten in ein komplett neues Gebiet gerade am Anfang (und danach auch immer wieder) ergeben. Danke auch für den Kontaktaufbau zu einigen der Koriphäen im Bereich der statistischen Netzwerkanalyse. Networking ist eben wichtig.

... my co-authors Alberto Caimo and Nial Friel. Thanks for the great time in Dublin, all the fruitful discussions, the code debugging, the help with the paper revisions, and especially the insights into Bayesian methods.

...Ernst Wit for the great workshop in Leiden on Statistical Network Science, and for agreeing to take the job as external examiner.

...meine langjährige Bürokollegin Linda Schulze Waltrup. Danke fürs Zuhören, wenn nötig in Ruhe lassen, fürs Pflanzen pflegen und fürs Sofa retten.

...meine Kollegen Verena Maier, Ludwig Bothmann und Michael Windmann. Danke für die entspannte Atmosphäre am Lehrstuhl, das Zuhören und das ab und zu Vorbeischauen, damit ich nicht vereinsame.

...Julia Kopf, Tina Feilke und Bernd Budich. Danke für die Hilfe beim Korrekturlesen der Arbeit, dass ihr mich zwischendurch immer wieder aufgebaut habt und vor allem danke für die langjährige Freundschaft.

...alle am Institut. Es herrscht wirklich eine unglaublich angenehme familiäre Atmosphäre, ohne die sowohl Studium, als auch Dissertation nur halb so schön gewesen wären. Danke dafür!

...meine Familie und Freunde, allen voran meine Eltern Karin Leon und Berthold Thiemichen. Danke, dass ihr mich immer unterstützt und dabei nicht zu oft nachgefragt habt, wie es denn so läuft.

...mein Mann Lorenz Thoelen. Danke, dass du da bist.

Zusammenfassung

Die Analyse von Netzwerkdaten gewinnt als Gebiet in der Statistik zunehmend an Bedeutung. Dabei sind sowohl die Modellierung selbst, als auch die damit verbundenen computationalen Aspekte herausfordernd. Sogenannte Exponential Random Graph Modelle (ERGM) sind ein bekannter und häufig genutzter Modellierungsansatz für solche Daten.

Diese Arbeit gibt zunächst eine kurze generelle Einführung in Exponential Random Graph Modelle zur statistischen Netzwerkanalyse. Vorteile und Probleme dieser Modellklasse werden diskutiert, wobei speziell das Problem der Modelldegeneriertheit beleuchtet wird. Die Dissertation beinhaltet zwei Beiträge zur Erweiterung von Exponential Random Graph Modellen. Die erste Erweiterung beruht auf einem Bayesianischen Ansatz. Dabei soll es ermöglicht werden, knoten-spezifische zufällige Effekte ins Modell aufzunehmen, um Heterogenität in den Knoten zu kompensieren. Der Ansatz basiert auf den von Caimo und Friel (2011) entwickelten Methoden für Bayesianische Exponential Random Graph Modelle. Zusätzlich zur Prozedur für die Modellanpassung wird eine approximative, aber praktikable Berechnung für Bayesfaktoren zur Modellselektion entwickelt. Die zweite Erweiterung nutzt einen Stichprobenansatz basierend auf der konditionalen Unabhängigkeitsstruktur eines Netzwerkes (wenn Markov-Unabhängigkeit angenommen wird) und erlaubt es große Netzwerke mit mehr als 1000 Knoten zu analysieren. Nicht-lineare Statistiken in Kombination mit einem penalisierten Glättungsansatz werden mit ins Modell aufgenommen, um das resultierende Modell zu verbessern und zu stabilisieren.

Beide Ansätze werden durch Datenbeispiele und im Falle des Bayesianischen Ansatzes durch eine Simulationsstudie illustriert. Alle entwickelten Methoden sind mittels der Open-Source Statistiksoftware R implementiert. Die Bayesianischen Methoden werden Teil des Pakets `Bergm` (Caimo und Friel, 2014). Stichproben- und Glättungsansatz sind im separaten Paket `ergm` implementiert und auf github verfügbar.

Summary

The analysis of network data is an emerging field in statistics which is challenging both model-wise and computationally. The so-called Exponential Random Graph Model (ERGM) is one particular well-known and widely used modelling approach for such data. This thesis starts with a short general introduction to Exponential Random Graph Models for statistical network analysis. Advantages and problems of this model class are presented with a special emphasis on the issue of model degeneracy. This dissertation contains two contributions which try to extend the existing Exponential Random Graph Models. The first extension uses a Bayesian approach in order to incorporate nodal random effects into the model to compensate for heterogeneity in the nodes of a network. It is based on the methodology developed by Caimo and Friel (2011) for Bayesian Exponential Random Graph Models. In addition to the model fitting procedure, an approximate but feasible calculation of the Bayes factor for model selection is developed. The second extension is based on a subsampling approach which utilizes the conditional independence structure of a network (if Markov independence is assumed) and allows to analyse larger networks with more than a thousand nodes. Non-linear statistics are added to the model using a penalized smoothing approach in order to improve and stabilise the resulting model. Both approaches are illustrated with concrete data examples, and in case of the Bayesian approach with a small simulation study. All developed methods are implemented using the open source statistics software **R**. The Bayesian methods will be made available in the package **Bergm** (Caimo and Friel, 2014). The smoothing approach is implemented in the separate add-on package **ergam** and available on github.

Contents

1	Introduction	1
1.1	Data Examples	3
1.2	Contributed Manuscripts and Software	5
1.2.1	Contributed Manuscripts	5
1.2.2	Software	6
2	Exponential Random Graph Models	7
2.1	From Random Graph to Exponential Random Graph Models	7
2.2	Properties of Exponential Random Graph Models	8
2.3	Estimation of Exponential Random Graph Models	11
2.3.1	Overview of Estimation Routines	11
2.3.2	Network Simulation	13
2.3.3	Goodness-of-Fit	14
2.4	Challenges and Solutions	15
2.4.1	Degeneracy	15
2.4.2	Geometrically Weighted Statistics	17
2.4.3	Nodal Heterogeneity	18
2.4.4	Further Limitations	19
3	Bayesian Exponential Random Graph Models with Nodal Random Effects	21
3.1	Introduction	23
3.2	Bayesian Model Formulation and Estimation	26
3.3	Model Selection	29
3.3.1	Bayesian Model Selection	29
3.3.2	Bayes Factor for Nested Models	30
3.3.3	Bayes Factor for Non-Nested Models	32
3.4	Examples	34
3.4.1	Data Examples	34
3.4.2	Simulation	50
3.5	Discussion and Summary	54

4	Stable Exponential Random Graph Models with Non-parametric Components for Large Dense Networks	57
4.1	Introduction	59
4.2	Estimation through Subsampling	62
4.3	Non-parametric Exponential Random Graph Models	65
4.3.1	Spline-Based Model	65
4.3.2	Penalized Estimation	66
4.3.3	Combining the Sample Estimates	70
4.4	Data Example	70
4.4.1	Linear Estimation through Subsampling	70
4.4.2	Non-parametric Estimation through Subsampling	72
4.5	Discussion	80
5	Further Ideas	83
5.1	Speed-up Bayesian Approach	83
5.2	Parallel Computing	84
5.3	Pseudo-Likelihood Bootstrap	85
6	The Bottom Line	87
	Appendices	III
A	Network Statistics	V
A.1	Some Examples with Notation and Formulae	V
A.2	Illustration of Markov Independence	VIII
B	R Code – Near-Degeneracy Illustration	XI
C	Laplace Approximation	XIII
	List of Figures	XVI
	List of Tables	XVII
	References	XIX

1 Introduction

“It is strange that the assumption that data obtained from human respondents represents independent replications has been so pervasive in statistical models used in sociological research.”

Tom A. B. Snijders (Snijders, 2016)

The field of network data analysis has gained more and more interest in recent years. In addition to the fact that a lot of network datasets are available, there is a wide range of possible applications, which often have been the driving forces behind method development in this field. Ranging from biological networks, such as genetic or metabolic pathways or gene regulatory networks, over technical and communication networks, to economic (e.g., company cooperations or trade flows) and – of course – social networks, the main feature of all of these data collections is its relational structure. Referring to Snijders quote at the beginning: Especially – but not only – in the sociological context, ignoring the relations and the thereby induced dependence structures among actors in an analysis may result in questionable conclusions. Dealing with relational structures and its dependencies is very challenging from a statistical point of view. Almost every basic statistics course contains the sentence “We assume the observations to be independent and identically distributed (*i.i.d.*)”, and the case of having observations which are not (conditionally) independent is not even considered in a lot of statistical curricula, especially if its only taught as a minor subject. The assumption of dealing with (at least conditionally) independent observations is almost always violated in the network context. This impedes, or even prevents the use of a lot of standard statistical models, as ignoring the interdependencies usually flaws the whole analysis.

Kolaczyk (2009) gives a gentle and comprehensive introduction to the field of statistical network analysis. The survey articles of Goldenberg et al. (2010), Hunter et al. (2012), Fienberg (2012), and Salter-Townshend et al. (2012), discuss recent statistical approaches, challenges, and developments in this domain.

This work focuses on a specific class of network models, so-called Exponential Random Graph Models (ERGM), which are rather well-developed, appealing for statisticians (as are exponential families in general), and therefore widely used. Lusher et al. (2013) give a general introduction to Exponential Random Graph Models.

The emphasis of this thesis lies on the modelling of network data consisting of nodes (or vertices, or actors), and edges (or ties, connections, relationships, or links) between them at one certain point in time. We do not include or consider information of the network over time, in the sense that we are only dealing with a single snapshot / observation of a network. A pair of nodes is denoted as a dyad, and the term dyadic describes the relationship (edge present or absent) between them. We represent the network as a graph and denote the $n \times n$ dimensional adjacency matrix of the graph with \mathbf{Y} , where n is the number of nodes in the network. The matrix element $Y_{ij} = 1$, if an edge exists between node i and node j , and $Y_{ij} = 0$ otherwise, with $i, j \in \{1, \dots, n\}$ and $i \neq j$. Thus, there is no connection from a node to itself (so-called self-loops). For simplicity we assume undirected edges, that is $Y_{ij} = Y_{ji}$. This means, if node A is connected to node B, node B is also connected to node A. Friendship networks usually yield an example of such undirected relationships (e.g., friends on Facebook). A classical example for directed relationships is Twitter, where users follow other users, but the relationship is not necessarily mutual. In case of an undirected graph the adjacency matrix is symmetric. For simplicity it is therefore sufficient to consider the upper triangle of \mathbf{Y} only, that is $Y_{ij}, j > i$, with $i, j \in \{1, \dots, n\}$. A lot of approaches for undirected graphs are also applicable to directed graphs. A concrete realisation of \mathbf{Y} is denoted with \mathbf{y} .

Covariates can be available for nodes (e.g., gender) or dyad-wise (e.g., indicators for same gender, or whether the actors went to the same school). Note that the interest lies in the link variables Y_{ij} , for $i, j \in \{1, \dots, n\}$, which are treated as random variables, while the nodes $i = 1, \dots, n$ are fixed.

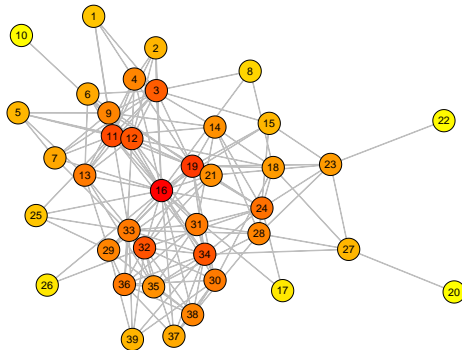
The concrete data examples employed in this thesis all belong to the field of social or political networks and are described in the next section. Nevertheless, the methods described in this thesis are not restricted to these fields of applications.

By using the graph representation of a network we may lose some information available in the data, and of course one needs to be careful when deciding what are the nodes, and especially what are the edges, when is an edge considered as being present and when as absent. All of this is discussed, e.g., in the chapter on “Mapping Networks” of Kolaczyk (2009). In this work the terms network and network graph are mainly used interchangeably. There is an ongoing discussion in the field of network analysis concerning the specification of the number of observations in a network, which also addresses definitions of asymptotic properties and more, see, e.g., Krivitsky and Kolaczyk (2015). As mentioned before, on scale of the whole network \mathbf{Y} we usually have a single observation \mathbf{y} . Another possibility is to consider the number of vertices n , which can also be regarded as network size. The size of a network is sometimes also defined through the number of edges n_E in the network, i.e. the edge variables with status $Y_{ij} = 1$. Our focus lies in the edge variables Y_{ij} themselves, whereas $\binom{n}{2} = \frac{n(n-1)}{2}$ of them are available. We do not want to restrict ourselves to one of the just mentioned concepts. It is nonetheless important to keep these aspects in mind when dealing with network data analysis.

1.1 Data Examples

In general, most examples used in this work deal with friendship networks to illustrate and exemplify matters.

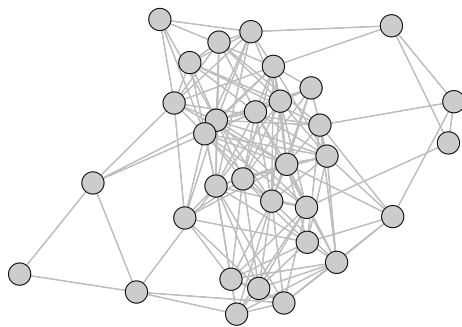




Kapferer taylor shop (Kapferer, 1972)

No. of nodes:	39
No. of existing edges:	158
No. of possible edges:	741
Network density:	0.21

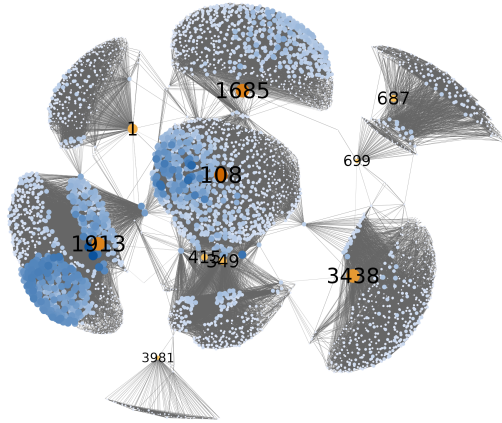
Interactions among workers in a tailor shop in Zambia.



European parliament members (Thurner et al., 2013)

No. of nodes:	32
No. of existing edges:	161
No. of possible edges:	496
Network density:	0.32

Induced subgraph from the Dutch members of the European parliament (MEP) in the 6th legislative period (the whole network consists of 900 vertices). Two MEPs are connected if they have at least one committee membership in common.



Facebook

(McAuley and Leskovec, 2012)

No. of nodes:	4,039
No. of existing edges:	88,234
No. of possible edges:	8,154,741
Network density:	0.01

Combined Facebook friendship data from ten ego-networks (an ego-net is the network among friends of a certain actor who is the ego, the ego itself is by definition connected to every other vertex in his ego-net).

1.2 Contributed Manuscripts and Software

1.2.1 Contributed Manuscripts

This thesis contains parts which are already published as stand-alone articles or available as online pre-prints, and which are joint work with co-authors. For the Bayesian approach in Chapter 3 this is

Thiemichen, S., Friel, N., Caimo, A., and Kauermann, G. (2016). Bayesian exponential random graph models with nodal random effects.
Social Networks, 46:11–28.

The subsampling and the non-parametric extension in Chapter 4 corresponds to

Thiemichen, S. and Kauermann, G. (2016). Stable exponential random graph models with non-parametric components for large dense networks.
arXiv preprint arXiv:1604.04732.

Information on the individual contributions of all involved authors can be found at the beginning of the respective chapters.

1.2.2 Software

All computations and most figures in this work have been produced using the **R** system for statistical computing (R Core Team, 2016), version 3.3.0 (if not stated otherwise), with packages **ergm** 3.6.0, **network** 1.13.0, and **statnet.common** 3.3.0. Information on additional add-on packages and our implemented routines is included at the beginning of the corresponding chapters. Our algorithms developed for model fitting and model selection in the Bayesian context in Chapter 3 will be included in the **Bergm** package (Caimo and Friel, 2014). The non-parametric approach together with the subsampling scheme in Chapter 4 is available in the package **ergam** on github (<https://github.com/sthiemichen/ergam>). All of our implementations are open-source software and therefore free to use for researchers and other users.

For the illustrative graphics in Chapter 2 and the appendix, and the model overview in Chapter 3 Inkscape (The Inkscape Team, 2015) has been used. The visualisation of the Facebook network data example in Chapter 4 has been generated using visone (Brandes and Wagner, 2004).

The picture on the title page shows the author's personal Facebook friendship network (as of 11th June 2012). The plot has been created using gephi (Bastian et al., 2009). The size of each vertex is proportional to the corresponding nodal degree.

2 Exponential Random Graph Models

2.1 From Random Graph to Exponential Random Graph Models

Various stochastic approaches have been proposed to model the adjacency matrix \mathbf{Y} of a network consisting of n nodes. One of the earliest and simplest models is the one suggested by Gilbert (1959). It is usually known as Bernoulli Random Graph Model. The model has the following form for the probability that an edge exists between node i and node j :

$$\mathbb{P}(Y_{ij} = 1) = \pi, \quad \forall i, j \in \{1, \dots, n\}, \text{ and } j > i, \quad (2.1)$$

with $\pi \in (0, 1)$. This model can be seen as a baseline or intercept-only model. For a large number of nodes n this formulation is equivalent to the model of Erdős and Rényi (1959), where all possible graphs for a given number of nodes n and a given number of existing edges n_E get the same uniform probability.

Extending this formulation to more realistic models lead to the so-called p_1 model (Holland and Leinhardt, 1981). It can be written as

$$\text{logit}[\mathbb{P}(Y_{ij} = 1)] = \log \left\{ \frac{\mathbb{P}(Y_{ij} = 1)}{1 - \mathbb{P}(Y_{ij} = 1)} \right\} = \alpha_i + \alpha_j + \mathbf{z}_{ij}^t \boldsymbol{\beta}, \quad (2.2)$$

where \mathbf{z}_{ij} denotes a set of covariates relating to the vertices i and j , and α_i and α_j are fixed nodal effects. The difference in the p_2 model (van Duijn et al., 2004; Zijlstra et al., 2006)

$$\begin{aligned} \text{logit}[\mathbb{P}(Y_{ij} = 1 | \boldsymbol{\phi})] &= \phi_i + \phi_j + \mathbf{z}_{ij}^t \boldsymbol{\beta}, \\ \boldsymbol{\phi} = (\phi_1, \dots, \phi_n)^t &\sim N(0, \sigma_\phi^2 I_n) \end{aligned} \quad (2.3)$$

with I_n as n dimensional unit matrix, is the treatment of the node-specific effects as nodal random effects ϕ_i , $i = 1, \dots, n$, which are considered to be latent and non-observable. This allows to tackle the problem of an increasing number of parameters with increasing number of nodes n in a rather elegant manner, as in most cases the nodal effects themselves are not of interest but only their variance parameter σ_ϕ^2 . In both models, the vertices are not

treated as homogeneous but as heterogeneous.

One of the main advantages of all previously mentioned models is that they fall into the classical generalized linear (mixed) model framework with binomial response and logit link function. Estimation can be carried out with available standard statistical software. When dealing with the p_2 model, a Bayesian approach can be used, as in a Bayesian framework including the additional normal distribution assumption needed for the random effects ϕ_i , $i = 1, \dots, n$, is straightforward via a prior distribution, see, e.g., Gill and Swartz (2004).

The class of so-called stochastic block models (see, e.g., Snijders and Nowicki, 1997; Nowicki and Snijders, 2001) are another extension of the Bernoulli model (2.1) and similar to the p_2 model. In these models, vertices are assigned to latent groups, which are also called blocks. Vertices belonging to the same block get the same probability for forming a tie among themselves, that is usually higher than the probability assigned to vertices belonging to different blocks. The probabilities can vary from block to block, and the block structure is usually unknown and therefore needs to be estimated as well. There are various extensions available, e.g., for considering mixed-memberships of vertices, that is vertices can belong to more than one block (Airoldi et al., 2008). Stochastic block models are applicable to large networks (see, e.g., Daudin et al., 2008). The same is true for the p_2 model. The concept of latent variables in stochastic block models has also been generalized in so-called latent network models (see, e.g., Hoff et al., 2002; Krivitsky and Handcock, 2008).

In all models presented so far, the probability that an edge exists between nodes i and j does not depend on the network structure itself. Let us consider the concrete example of friendship networks. When trying to model the probability that two actors become friends with each other, considering the number of friends the two have in common is usually a reasonable thing to do (the phenomenon is known as triangulation). Including this additional covariate structure is possible in the Exponential Random Graph Model (ERGM), which is sometimes denoted as p^* (p -star) model, and has been proposed by Frank and Strauss (1986).

2.2 Properties of Exponential Random Graph Models

By employing Exponential Random Graph Models we use the likelihood function to model the network \mathbf{Y}

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \boldsymbol{\theta}) = f(\mathbf{y} | \boldsymbol{\theta}) = \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}}{\kappa(\boldsymbol{\theta})}, \quad (2.4)$$

where $\boldsymbol{\theta} = (\theta_0, \dots, \theta_p)^t$ denotes the vector of model parameters and $s(\mathbf{y}) = (s_0(\mathbf{y}), \dots, s_p(\mathbf{y}))^t$ is a vector of network statistics. It is easy to see that an exponential family is assumed. Therefore, the vector of network statistics $s(\mathbf{y})$ is sufficient, with $\boldsymbol{\theta}$ being the vector of natural or canonical parameters. For more details of exponential families and the associated properties see, e.g., Appendix A.1 of Fahrmeir and Tutz (2001). The vector $s(\mathbf{y})$ can contain counts of edges, 2-stars, triangles, cycles or other structures (sub-graphs) in the network, see, e.g., Snijders et al. (2006). Figure 2.1 illustrates some of these possible network sub-graphs for an undirected network. Section A.1 in the appendix contains further examples and the corresponding formulae. In addition, it is also possible to incorporate available covariate information like, for instance, counts of edges between actors having the same gender, or who went to the same school. For a comprehensive and a bit more technical overview on model specification of ERGMs see Morris et al. (2008).

When including only structural terms, like k -stars and triangles, the resulting Exponential Random Graph Model has a Markovian independence structure, that is two edges are conditionally independent, given the rest of the network, if they do not share a node (Frank and Strauss, 1986; Whittaker, 2009; Koskinen and Daraganova, 2013). Literally speaking, this results in a dyad depending only on its direct neighbouring dyads (i.e. they share at least one node). For ERGMs containing cycles (4-cycles or higher cycles), k -triangles, or other more complex structures this property does not hold, see Section 4.1, and Section A.2 in the appendix. For an overview of induced dependence structures depending on the specified ERGM we refer to Pattison and Snijders (2013), and the references therein.

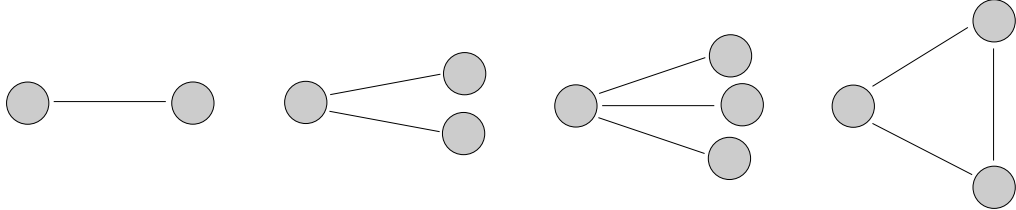


Figure 2.1 *Examples of network sub-graphs: Edge (1-star), 2-star, 3-star, and triangle.*

The term $\kappa(\boldsymbol{\theta})$ in (2.4) denotes the normalizing constant of the exponential family, that is

$$\kappa(\boldsymbol{\theta}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}.$$

It is accordingly the sum over $2^{\binom{n}{2}}$ potential undirected graphs. This number becomes infeasible to calculate even for rather small networks. For $n = 10$ we already have to deal with more than $3.5 \cdot 10^{13}$ terms. This constant is therefore usually numerically intractable. Coming from the properties of the exponential family type distribution one has for the log

normalizing constant

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log(\kappa(\boldsymbol{\theta})) = \mathbb{E}(s(\mathbf{Y})|\boldsymbol{\theta}) \quad (2.5)$$

and

$$\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^t} \log(\kappa(\boldsymbol{\theta})) = \text{Cov}(s(\mathbf{Y}), s(\mathbf{Y})^t | \boldsymbol{\theta}). \quad (2.6)$$

Both expressions can therefore be estimated based on a sample of graphs coming from this ERGM with parameters $\boldsymbol{\theta}$. More details on estimation and simulation are given in the next section.

Model (2.4) can be rewritten in a conditional form

$$\text{logit}[\mathbb{P}(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\theta})] = \boldsymbol{\theta}^t s_{ij}(\mathbf{y}), \quad (2.7)$$

where $s_{ij}(\mathbf{y})$ denotes the vector of so called change statistics

$$s_{ij}(\mathbf{y}) = \Delta_{ij}s(\mathbf{y}) = s(y_{ij} = 1, y_{kl}, (k, l) \neq (i, j)) - s(y_{ij} = 0, y_{kl}, (k, l) \neq (i, j)).$$

The change statistics reflect the difference in counts of edges, 2-stars, triangles and so on, which occur between toggling the edge y_{ij} from existent to non-existent given the rest of the network. This formulation allows for a rather straightforward conditional interpretation focusing on a single edge Y_{ij} . Some statistics are easy to interpret especially using this conditional framework. For instance, if we include the number of triangles in the model, the triangle change corresponds to the number of common friends node i and j have, given the rest of the network, and the parameter $\theta_{\text{triangles}}$ is accordingly the linear effect of one additional common friend on the log-odds of becoming friends, keeping all other covariates constant. Note that some structural covariates are much harder to interpret. We will give some examples in Section 2.4.

This conditional formulation of the ERGM in (2.7) also shows the similarity to the p_1 and p_2 formulations in (2.2) and (2.3), respectively. Note that, as we are including structural network terms, the single observations Y_{ij} are not (conditionally) independent here. There is an exception, of course, if we employ an intercept-only model. If we only include the total edge count, the edge change is always one per edge variable and the linear predictor in (2.7) consists only of θ_0 , which captures the overall tendency of any dyad of forming a tie in the network. In this case the model is equivalent to the Bernoulli Random Graph Model in (2.1).

For a more detailed discussion of properties of Exponential Random Graph Models, we refer to Robins et al. (2007a), Robins et al. (2007b), and Lusher et al. (2013).

2.3 Estimation of Exponential Random Graph Models

2.3.1 Overview of Estimation Routines

There is a variety of estimation approaches available in the context of Exponential Random Graph Models. The following descriptions are by far not exhaustive. In the frame of this work, only a general overview and a sketch of the most prominent approaches are provided. For a deeper discussion of the algorithms we refer to Hunter et al. (2012), and Koskinen and Snijders (2013).

Pseudo-Likelihood Estimation

One of the simplest approaches for estimation of model (2.4) is the Pseudo-Likelihood estimation (Strauss and Ikeda, 1990). The conditional probability in (2.7) is evaluated for each edge variable Y_{ij} , for $i, j \in \{1, \dots, n\}$ and $j > i$, given the rest of the network. The product of the conditional probabilities

$$\prod_{\substack{i,j \in \{1, \dots, n\} \\ j > i}} \mathbb{P}(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\theta})$$

is used as an approximation of the likelihood. In other words, the dependence structure of the link variables is ignored and the estimation is carried out similar to every standard logistic regression model. Note that the Maximum Pseudo-Likelihood Estimator (MPLE) is an unbiased estimator if the link variables are conditionally independent, which is the case in the p_1 , p_2 , and Bernoulli framework, where no structural network effects are considered. Lubbers and Snijders (2007), as well as van Duijn et al. (2009) have shown that the MPLE is biased in most scenarios. Even though there are approaches to correct for this bias using the Fischer information (see Firth, 1993), there is a loss in efficiency compared to the maximum likelihood estimator.

Stochastic Approximation (Robbins-Monro)

As exact computation of the normalizing constant $\kappa(\boldsymbol{\theta})$ in (2.4) is infeasible in almost all real data applications, one way to tackle the estimation problem are simulation based methods, which are usually quite demanding from a computational point of view. Snijders (2002) suggests the use of stochastic approximation (Robbins and Monro, 1951) in the calculation of $\partial \kappa(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ in the score equation resulting from (2.4) and making use of property (2.5) when computing the MLE. To do so, a Markov Chain Monte Carlo (MCMC) based sample $\mathbf{y}^{(t)}$ from an ERGM is required in every iterative update step t , consisting

only of a single simulated graph based on the current estimate of $\boldsymbol{\theta}^{(t)}$. For the iterative procedure an initial guess for the value of $\boldsymbol{\theta}$ is needed. This starting value $\boldsymbol{\theta}^{(0)}$ can be obtained using, for instance, the MPLE. The iterative updating from $\boldsymbol{\theta}^{(t)}$ to $\boldsymbol{\theta}^{(t+1)}$ is done using a Newton-Raphson-type procedure, that is

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - a_{(t)} \mathbf{D}_0^{-1} (s(\mathbf{y}^{(t)}) - s(\mathbf{y})),$$

where \mathbf{D}_0 is a scaling matrix. \mathbf{D}_0 is determined in an initialisation phase as the diagonal of an estimated covariance matrix of the sufficient statistics based on some graph samples (compare to equation (2.6)) using $\boldsymbol{\theta}^{(0)}$. The term $a_{(t)}$ denotes a step length between zero and one, which becomes smaller in each iteration and is needed to avoid “overshooting” of the MLE. For more details on the stochastic approximation approach see, e.g., Koskinen and Snijders (2013).

MCMC MLE (Geyer-Thompson)

Handcock (2003), and later Hunter and Handcock (2006) adopted an alternative for the computation of the maximum likelihood estimator in ERGMs, the MCMC MLE of Geyer and Thompson (1992). This approach is sometimes referred to as “importance sampling”. Instead of maximizing the likelihood directly, the likelihood ratio is considered, which results in the log-likelihood formulation

$$\ell(\boldsymbol{\theta}) - \ell(\boldsymbol{\theta}_0) = (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^t s(\mathbf{y}) - \log \left\{ \frac{\kappa(\boldsymbol{\theta})}{\kappa(\boldsymbol{\theta}_0)} \right\}. \quad (2.8)$$

Again, by default, the easy to calculate MPLE can be used as a starting value $\boldsymbol{\theta}_0$ to generate an initial sample of graphs from an ERGM based on the value $\boldsymbol{\theta}_0$. Then, the ratio of normalizing constants

$$\frac{\kappa(\boldsymbol{\theta})}{\kappa(\boldsymbol{\theta}_0)} = \mathbb{E} \left(\exp \left\{ (\boldsymbol{\theta} - \boldsymbol{\theta}_0)^t s(\mathbf{Y}) | \boldsymbol{\theta}_0 \right\} \right)$$

is approximated based on this sample. This allows to maximize the log-likelihood (2.8) as a function of the parameters $\boldsymbol{\theta}$, which are of interest. The goal is to get an expected value of the statistics $\mathbb{E}(s(\mathbf{Y}))$, which is approximated using the initial sample of graphs, as close as possible to the observed value of the statistics $s(\mathbf{y})$ by iteratively updating the estimate of $\boldsymbol{\theta}$ to $\boldsymbol{\theta}^{(t+1)}$, again, using a Newton-Raphson-type equation. Note that in each step only the initial sample of graphs is used. Problems usually arise when the initial value of $\boldsymbol{\theta}_0$ is not close enough to the true parameter vector $\boldsymbol{\theta}$, sometimes even resulting in an expression which cannot be maximized at all, see, e.g., Figure 2 of Hunter et al. (2008b), or Figure

3 of Caimo and Friel (2011). The algorithm can then be restarted (including new initial simulations of graphs) with other initial values, obtained either as the last estimate from the previous run, or completely new ones specified by the user. Nevertheless, approximation of a ratio of such normalizing constants is in general much more stable than approximating one of them directly.

Other Approaches

Caimo and Friel (2011) propose a Bayesian framework which is based on the exchange algorithm of Murray et al. (2006) and circumvents the calculation of normalizing constants completely. This approach is described in more detail in Section 3.2. Again, simulations of networks are needed to employ this strategy. A common method for network simulation from an ERGM is described in the next subsection.

The stochastic approximation approach (Robbins-Monro algorithm) is less prone to bad initial values than the MCMC MLE (Geyer-Thompson algorithm). Therefore, in general, it is quite robust, even though it is less efficient than the MCMC MLE, where only an initial sample of graphs is needed. The non-Bayesian ERGM estimation approaches described so far are, e.g., available in the R package `ergm` (Hunter et al., 2008b), which is also part of the `statnet` suite of packages (Handcock et al., 2008).

There are several extensions and suggestions for improvements of estimation in the ERGM framework available, e.g., by Hummel et al. (2012). For a more detailed description of available estimation routines and extensions we refer to the survey article of Hunter et al. (2012), the book of Lusher et al. (2013), and the references therein.

2.3.2 Network Simulation

Even though the normalizing constant of the Exponential Random Graph Model in (2.4) is in general intractable, simulating networks from an ERGM is possible and relatively straightforward to implement. A common approach for simulating networks from a specific Exponential Random Graph Models with parameter vector θ^* is the so-called “tie no tie” (TNT) sampler, as available in the `ergm` package (Hunter et al., 2008b). The R code in Appendix B contains an example for the simulation functionality in this package. The TNT approach usually starts from an empty network, i.e. the $n \times n$ adjacency matrix contains only entries of zero. In each step, a single edge variable Y_{ij} is selected at random and toggled, i.e. set to one, if currently zero, or set to zero, if currently one. This results in

$\mathbf{y}_{\text{proposed}}$. The toggle is accepted with probability

$$\alpha = \min \left\{ 1, \frac{\mathbb{P}(\mathbf{Y} = \mathbf{y}_{\text{proposed}} | \boldsymbol{\theta}^*)}{\mathbb{P}(\mathbf{Y} = \mathbf{y}_{\text{current}} | \boldsymbol{\theta}^*)} \right\},$$

where the needed probabilities result from the ERGM with parameter vector $\boldsymbol{\theta}^*$. Note that in the probability ratio in α the normalizing constants simply cancel out, as they are identical for both probabilities $\mathbb{P}(\mathbf{Y} = \mathbf{y}_{\text{proposed}} | \boldsymbol{\theta}^*)$ and $\mathbb{P}(\mathbf{Y} = \mathbf{y}_{\text{current}} | \boldsymbol{\theta}^*)$. The TNT in the `ergm` package does not select edge variables uniformly at random, but uses a probability of 0.5 to select an edge variable which is currently zero. This normally leads to faster convergence of the Markov chain, which is more effective in sparse settings, as less iterations are needed for simulating a single network, see Morris et al. (2008), and Hunter et al. (2008b) for details.

Often, if several simulated networks based on the same parameters are needed, a single Markov chain is used, allowing the chain to burn-in (the burn-in is the number of iterations before the first network is stored from the chain) and using a smaller number of interval iterations between the successive draws. This is usually more efficient than employing several chains, each starting from scratch, and each of them providing a single simulated network only. The resulting draws are correlated in this case. There are some diagnostic tools and theoretical results available concerning the number of iterations needed to obtain a draw from the target distribution when simulating a network. The chain generating the network samples needs a sufficiently large burn-in and the number of iterations between each successively generated network using this chain should be large enough as well. Checking the autocorrelation of successive draws is an option to see if enough interval iterations are used, see, e.g., Koskinen and Snijders (2013). The autocorrelation should not be too high. Looking at trace plots of counts of network statistics (e.g., edges, 2-stars, or triangles) can help to identify a sufficiently large burn-in. Following Everitt (2012), in the Bayesian framework (Caimo and Friel, 2011) it is often sufficient to use roughly the number of possible ties in the network as number of iterations of the Markov chain to obtain a single network. In general, the extensive number of simulations needed in estimation of ERGMs is one of the main causes of the computational burden associated with this model class.

2.3.3 Goodness-of-Fit

Goodness-of-fit (GOF) assessment of Exponential Random Graph Models is usually obtained based on samples of graphs simulated from the fitted model. Hunter et al. (2008a) proposed to consider network statistics like, e.g., the distributions of degree, edge-wise-shared partners (for existing edges the number of common neighbours of the two vertices involved), or geodesic distance (the shortest path length between two nodes), and compare

them graphically between the sampled networks and the original network (see also Robins et al., 2007b). The statistics do not necessarily have to be included as model terms in the fitted ERGM. If the observed values fall in the range of the simulated ones, it is an indicator that the model is suitable for these data aspects. If this is not the case, the model is not capable of capturing some data properties. However, there is often no direct indication why the model performs badly and how to proceed in order to improve model fit. We employ this classical GOF procedure in the Bayesian framework in Section 3.4.1. As can be seen here, this approach for assessing model fit follows different routes than traditional likelihood based inference, such as likelihood ratio tests, etc.. This is due to the fact that the likelihood itself is difficult to evaluate because of the aforementioned problems implied by the unknown normalizing constant. For an overview of more classical type testing routines in ERGMs, see Koskinen and Snijders (2013).

In Chapter 4, we employ Pearson residuals as an additional diagnostic tool, which allows us to identify at least some potential problems causing difficulties in model fit.

2.4 Challenges and Solutions

2.4.1 Degeneracy

Apart from the already mentioned computational drawbacks for MLE estimation in the Exponential Random Graph framework due to extensive MCMC based routines, the model class suffers from so-called degeneracy problems, see, for instance, Snijders et al. (2006), Schweinberger (2011), and Chatterjee and Diaconis (2013). This means that for a lot of parameter values θ , most of the probability mass is placed on configurations yielding completely empty or completely full graphs. As pointed out before, the maximum likelihood estimation routines are simulation based and hence, generating only empty or full graphs is problematic. The remaining parameter space resulting in reasonable networks is often peculiarly shaped and small, see, e.g., Handcock (2003), Rinaldo et al. (2009), and Schweinberger (2011). A lot of basic models which are usually appealing because of the easy to interpret effects, e.g., the ones containing only 2-stars, or higher k -stars, or triangles, show near-degenerate behaviour. To illustrate the issue we simulate networks with $n = 30$ nodes from an Exponential Random Graph Model with edge and 2-star statistics, i.e.

$$\mathbb{P}(Y = y|\theta) = \frac{\exp\{\theta_{\text{edge}} s_{\text{edge}}(y) + \theta_{2\text{-star}} s_{2\text{-star}}(y)\}}{\kappa(\theta)}.$$

The parameter θ_{edge} is constantly set to a value of -2 , and the parameter associated with the 2-star effect is varied between -1 and 1 . For each value, we simulate 50 networks and Figure 2.2 shows the average density of the resulting networks. The density of a

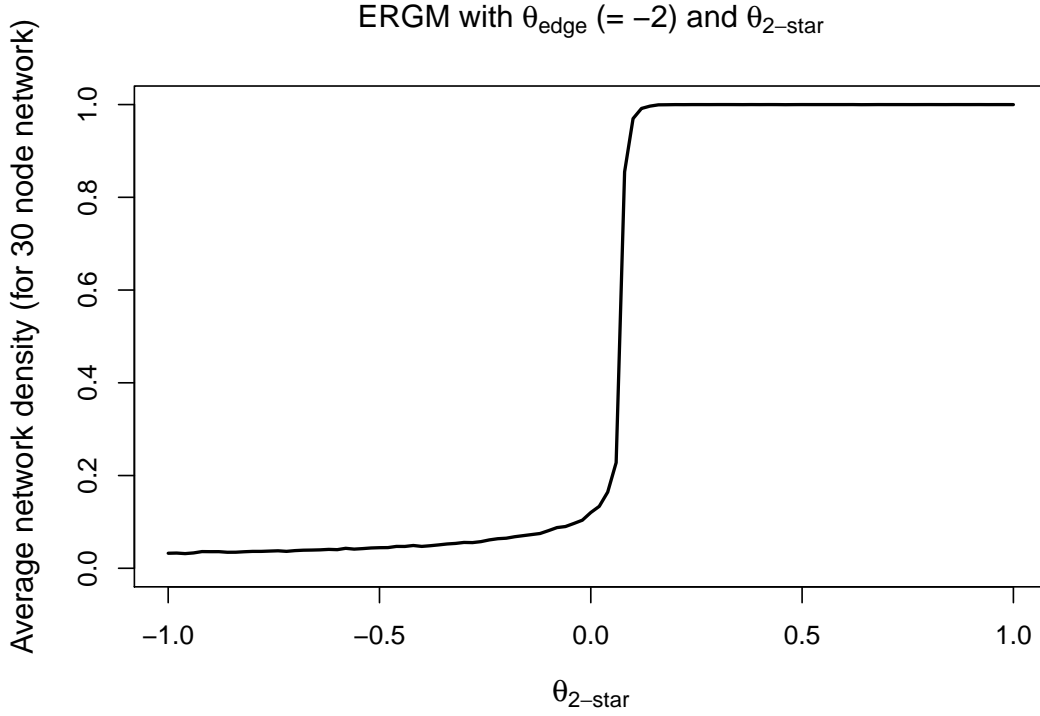


Figure 2.2 Resulting average network density when simulating graphs with $n = 30$ nodes from an ERGM containing edge and 2-star counts.

network is calculated as the number of existing edges divided by the number of possible edges $d(\mathbf{y}) = n_E / \binom{n(n-1)}{2}$. The corresponding R code for the simulation can be found in Appendix B. The sharp transition from a rather sparse to a full graph for only a small change of the value of $\theta_{2\text{-star}}$ is apparent. This transition becomes even sharper with increasing n , which results in the region of parameter values yielding reasonable graphs becoming even smaller.

For an ERGM containing edges, 2-stars, and triangles as sufficient statistics, Schweinberger (2011) has shown that only a linear combination of the form

$$\theta_{2\text{-star}} = -\frac{\theta_{\text{triangle}}}{3} \quad (2.9)$$

results in a stable setting, that is the obtained graphs are not completely full or completely empty, independently of the value of n . This result is in line with the often stated practical recommendation of compensating a positive triangle effect with a small negative 2-star effect. Figure 2.3 illustrates the stable setting in (2.9). We see that there still is a transition, but clearly less sharp and not resulting in completely full graphs. When assuming this setting for model estimation, only one parameter remains to be estimated. When doing so,

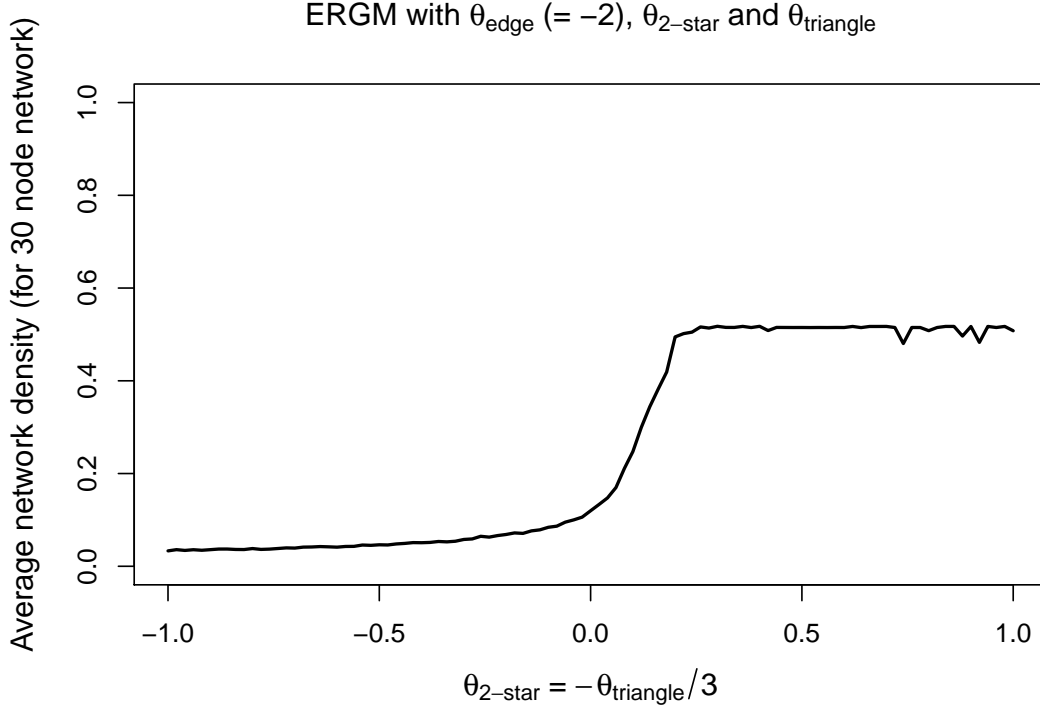


Figure 2.3 Resulting average network density when simulating graphs with $n = 30$ nodes from an ERGM containing edge, 2-star, and triangle counts with $\theta_{2\text{-star}} = -\frac{\theta_{\text{triangle}}}{3}$.

the resulting combined statistic¹ is not straightforward to interpret.

Caimo and Friel (2011) have shown that the Bayesian Exponential Random Graph Model behaves more stable in near-degenerate situations, where estimation in the standard ERGM already breaks down.

2.4.2 Geometrically Weighted Statistics

The instability appearing in Exponential Random Graph Models has led to the proposal of modified statistics as, for instance, alternating star or alternating triangle statistics (Snijders et al., 2006). From a modelling point of view, the geometrically weighted degree (GWD) statistic

$$e^{\theta_{\text{dec}}} \sum_{i=1}^{n-1} \left\{ 1 - \left(1 - e^{-\theta_{\text{dec}}} \right)^i \right\} D_i(\mathbf{y}),$$

¹ When restricting an ERGM containing edges, 2-stars, and triangles to setting (2.9), we obtain

$$\theta_{2\text{-star}} s_{2\text{-star}}(\mathbf{y}) - 3 \theta_{2\text{-star}} s_{\text{triangle}}(\mathbf{y}) = \theta_{2\text{-star}} s_{\text{combined}}(\mathbf{y}),$$

with $s_{\text{combined}}(\mathbf{y}) = s_{2\text{-star}}(\mathbf{y}) - 3 s_{\text{triangle}}(\mathbf{y})$.

where $D_i(\mathbf{y})$ denotes the number of nodes with degree i , and the geometrically weighted edge-wise shared partners (GWESP) statistic

$$e^{\theta_{\text{dec}}} \sum_{i=1}^{n-2} \left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^i \right\} \text{EP}_i(\mathbf{y}),$$

where $\text{EP}_i(\mathbf{y})$ denotes the number of edges with i shared partners, are equivalent to the alternating statistics (Hunter, 2007; Goodreau, 2007). Both statistics use an exponential down-weighting of the incorporated counts. The parameter θ_{dec} is usually set to some fixed value (the standard choice is $\log(2)$). In the framework of Curved Exponential Family Models (Hunter and Handcock, 2006), the parameter θ_{dec} is estimated as well. These types of statistics stabilise the whole model fitting, but are very difficult to interpret.

In Chapter 4, we propose an alternative by adding smooth functional components to the model, based on penalized estimation in the context of non-parametric models (see Ruppert et al., 2003), while maintaining the intuitive interpretability of statistics like 2-stars and triangles.

2.4.3 Nodal Heterogeneity

In ERGMs nodes are assumed to be homogeneous, except for differences captured in available (nodal) covariates. This assumption may be unrealistic, especially in the context of social networks, where some actors tend to attract a lot of connections, while others prefer to stay on their own. This difference can not necessarily be explained completely by covariates, like gender, age, etc.. The modelling approach of the p_2 model in (2.3) therefore yields the baseline for our first extension of the ERGM in Chapter 3 by adding nodal random effects to the model, resulting in

$$\text{logit} \left[\mathbb{P}(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\theta}, \phi_i, \phi_j) \right] = \boldsymbol{\theta}^t s_{ij}(\mathbf{y}) + \phi_i + \phi_j \quad (2.10)$$

with $\boldsymbol{\phi} = (\phi_1, \dots, \phi_n)^t$ and $\phi_i \stackrel{\text{i.i.d.}}{\sim} N(\mu_\phi, \sigma_\phi^2)$, $i = 1, \dots, n$. The extended model falls in the general class of Exponential-family Random Network Models, proposed by Fellows and Handcock (2012). Krivitsky et al. (2009) also develop a model with actor-specific random effects based on a latent cluster model. We follow this line of thought and add interpretability of the approach by considering the parameter σ_ϕ^2 as a measure of nodal heterogeneity. There is a connection between the nodal random effects ϕ_i , $i = 1, \dots, n$ and the nodal degrees, as these are the statistics associated with the random effects, which is explained in greater detail in Section 3.1. This interpretation is related to the work of Snijders (1981), where the author uses the degree variance as a measure of graph heterogeneity.

2.4.4 Further Limitations

Another issue arising when dealing with Exponential Random Graph Models is that these models are not consistent under sampling (Shalizi and Rinaldo, 2013). If we are interested in a population of, e.g., n^* nodes, but we only have data on the induced sub-graph of n nodes, where $n < n^*$, fitting an ERGM to the sub-graph does not give reasonable estimates for the whole network of n^* nodes. This whole issue comes back to the notion of n , the number of nodes, being fixed when fitting an Exponential Random Graph Model. It implies that parameter estimates resulting from ERGMs fitted to networks with a different number of nodes are not directly comparable. One should keep this in mind when interpreting the model outcomes. This problem limits the generalisability of results obtained from Exponential Random Graph Models to a greater population when only data on a sample was available for model fitting.

The issue of missing data in the network context is a topic of its own and not specific to Exponential Random Graph Models. However, there are a few approaches available in context of ERGMs to cope with edge variables being unobserved, i.e. when there is no information available whether these edges exist or not, see, e.g., the work of Handcock and Gile (2010) in the context of different network sampling designs.

A further limitation is the restriction to the modelling of binary tie variables only. This may cause a loss in information if the available data consists of weighted edges, like for example trade flows, where the amount is recorded and not only whether there was trade between two nodes or not. Krivitsky (2012), and Krivitsky and Butts (2015), or Desmarais and Cranmer (2012a), for instance, propose some strategies for dealing with such valued networks.

Classical Exponential Random Graph Models are static, meaning that they allow to model cross-sectional data. An assumption related to this property is that the network we see comes from its stationary distribution. This assumption can be violated especially if the underlying network-generating process has not settled in already. There are extensions available for modelling longitudinal network data, see, e.g., Snijders et al. (2010a). For a general overview of available approaches in or related to the Exponential Random Graph context we refer to Snijders and Koskinen (2013). Hanneke et al. (2010), for instance, have proposed time discrete models. Another widely used and well developed modelling approach for longitudinal network data are the stochastic-actor oriented models (SAOM; see, e.g., Snijders, 2001; Snijders et al., 2010b). They are not tie oriented as the ERGMs, but the main difference lies in the SAOM focusing on the changes occurring during two points in time. Consequently, stationarity of the underlying process is not assumed.

However, as the focus of this thesis lies in extending cross-sectional Exponential Random Graph Models, we are dealing with single observations of networks at a certain point in time, consisting of binary edge variables, and containing no missing data.

3 Bayesian Exponential Random Graph Models with Nodal Random Effects

Abstract

We extend the well-known and widely used Exponential Random Graph Model (ERGM) by including nodal random effects to compensate for heterogeneity in the nodes of a network. The Bayesian framework for ERGMs proposed by Caimo and Friel (2011) yields the basis of our modelling algorithm. A central problem in network models is the question of model selection and following the Bayesian paradigm we focus on estimating Bayes factors. To do so we develop an approximate but feasible calculation of the Bayes factor which allows to pursue model selection. Three data examples and a small simulation study illustrate our mixed model approach and the corresponding model selection.

Contributed Manuscript

This chapter is in most parts equivalent to the final submitted version of the publication

Thiemichen, S., Friel, N., Caimo, A., and Kauermann, G. (2016). Bayesian exponential random graph models with nodal random effects. *Social Networks*, 46:11–28.

except for a few corrections, mainly concerning orthography, and small adjustments as the paper serves as a chapter in this thesis and no longer as stand-alone article.

This is joint work with Nial Friel (School of Mathematical Sciences and Insight: The National Centre for Data Analytics, University College Dublin, Ireland), Alberto Caimo (School of Mathematical Sciences, Dublin Institute of Technology, Ireland), and Göran Kauermann (Institut für Statistik, Ludwigs-Maximilians-Universität München, Germany). The basic idea of including nodal random effects into the framework of Exponential Random Graph Models came from Göran Kauermann, and Stephanie Thiemichen. Nial Friel proposed to use Bayes factors for model selection. All authors contributed to the concrete development of the model extension, and application to data examples. Stephanie Thiemichen wrote the algorithmic implementation, based on code from Alberto Caimo in the R package `Bergm`, conducted the simulation study, and performed the data analysis.

Alberto Caimo helped with code debugging. Most of the manuscript was written by Stephanie Thiemichen, and Göran Kauermann. All authors contributed to the discussion section of the article, and were involved in proof-reading the manuscript.

Software

All computations and plots in this chapter have been produced using R version 3.2.2 with packages `Bergm` 3.0.1, `mvtnorm` 1.0-3, `coda` 0.18-1, `ergm` 3.5.1, `network` 1.13.0, and `statnet.common` 3.3.0. For parallelisation of the simulation and for the Bayes factor computation R's base package `parallel` was used, where possible.

Our algorithms for model fitting and model selection will be included in the `Bergm` package.

The graphical model overview in Figure 3.1 has been generated using Inkscape (version 0.48.4).

3.1 Introduction

The analysis of network data is an emerging field in statistics which is challenging both model-wise and computationally. Recently Goldenberg et al. (2010), Hunter et al. (2012), Fienberg (2012), and Salter-Townshend et al. (2012), respectively, published comprehensive survey articles discussing statistical approaches, challenges and developments in network data analysis. We also refer to the monograph of Kolaczyk (2009) for a comprehensive introduction to the field.

In this chapter we consider networks represented as a $n \times n$ dimensional adjacency matrix \mathbf{Y} , where the element $Y_{ij} = 1$, if an edge exists between vertex i and vertex j , and $Y_{ij} = 0$ otherwise, with $i, j \in \{1, \dots, n\}$ and $i \neq j$, that is there is no connection from a vertex to itself. With n we denote the number of vertices in the network and for simplicity we assume undirected edges, that is $Y_{ij} = Y_{ji}$. Therefore, the matrix \mathbf{Y} is symmetric and for simplicity it is sufficient to consider the upper triangle of \mathbf{Y} only, that is $Y_{ij}, j > i$. Our approach equally applies to non-symmetric adjacency matrices corresponding to directed graphs. A concrete realisation of \mathbf{Y} is denoted with \mathbf{y} .

With respect to the available statistical models for modelling cross-sectional network data one may roughly distinguish between two strands, (a) models which explain the existence of an edge purely with external nodal covariates or random effects and (b) models where the existence of an edge also depends on the local network structure. The first strand of models is phrased as p_1 and p_2 models tracing back to Holland and Leinhardt (1981). Specifically, in the p_1 model we set

$$\text{logit}[\mathbb{P}(Y_{ij} = 1)] = \log \left\{ \frac{\mathbb{P}(Y_{ij} = 1)}{1 - \mathbb{P}(Y_{ij} = 1)} \right\} = \alpha_i + \alpha_j + \mathbf{z}_{ij}^t \boldsymbol{\beta}, \quad (3.1)$$

where \mathbf{z}_{ij} denotes a set of covariates relating to the vertices i and j and α_i and α_j are nodal effects, here assuming undirected edges. Since the number of parameters increases with increasing network size n , van Duijn et al. (2004) proposed to replace the α parameters in (3.1) by random effects, see also Zijlstra et al. (2006). This yields the p_2 model

$$\begin{aligned} \text{logit}[\mathbb{P}(Y_{ij} = 1 | \boldsymbol{\phi})] &= \phi_i + \phi_j + \mathbf{z}_{ij}^t \boldsymbol{\beta}, \\ \boldsymbol{\phi} = (\phi_1, \dots, \phi_n)^t &\sim N(0, \sigma_\phi^2 I_n) \end{aligned} \quad (3.2)$$

with I_n as n dimensional unit matrix. A general principle with this approach is that vertices (or actors in the network, respectively) are not considered as homogeneous but heterogeneous, though their heterogeneity is not observable but latent and expressed in the node specific random effects ϕ_i , $i = 1, \dots, n$.

Both, the p_1 and the p_2 model lie within the classical generalized linear (mixed) model

framework which allows estimation using standard statistical software. The p_2 models also allow for Bayesian estimation approaches, see for example Gill and Swartz (2004).

The second strand in statistical network modelling is based on the so called Exponential Random Graph Model (ERGM) proposed by Frank and Strauss (1986). Here we model directly the network using the likelihood function

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}|\boldsymbol{\theta}) = f(\mathbf{y}|\boldsymbol{\theta}) = \frac{q_{\boldsymbol{\theta}}(\mathbf{y})}{\kappa(\boldsymbol{\theta})} = \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}}{\kappa(\boldsymbol{\theta})}, \quad (3.3)$$

where $\boldsymbol{\theta} = (\theta_0, \dots, \theta_p)^t$ is the vector of model parameters and $s(\mathbf{y}) = (s_0(\mathbf{y}), \dots, s_p(\mathbf{y}))^t$ is a vector of sufficient network statistics like the number of edges or 2-stars in a network, see for example Snijders et al. (2006). In equation (3.3) the term $\kappa(\boldsymbol{\theta})$ denotes the normalizing constant, that is

$$\kappa(\boldsymbol{\theta}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}$$

and is accordingly the sum over $2^{\binom{n}{2}}$ potential undirected graphs and therefore numerically intractable, except for very small graphs. Early fitting approaches are based on the Pseudo-Likelihood idea proposed by Strauss and Ikeda (1990). More advanced are MCMC based routines proposed by Hunter and Handcock (2006) based on the work of Geyer and Thompson (1992). A fully Bayesian approach to estimate ERGMs has been developed by Caimo and Friel (2011).

Model (3.3) allows for a conditional interpretation by focusing on the occurrence of a single edge between two nodes. To be specific we obtain

$$\text{logit}\left[\mathbb{P}(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\theta})\right] = \boldsymbol{\theta}^t s_{ij}(\mathbf{y}), \quad (3.4)$$

where $s_{ij}(\mathbf{y})$ denotes the vector of so called change statistics

$$s_{ij}(\mathbf{y}) = s(y_{ij} = 1, y_{kl}, (k, l) \neq (i, j)) - s(y_{ij} = 0, y_{kl}, (k, l) \neq (i, j)).$$

We refer to Robins et al. (2007a), Robins et al. (2007b), and the rather recent work of Lusher et al. (2013) for a deeper discussion of Exponential Random Graph Models.

Contrasting equation (3.4) with the p_1 and p_2 model given in equations (3.1) and (3.2) it becomes obvious that the ERGM in contrast to the p_1 and p_2 models take the network structure into account while considering the nodes to be homogeneous. When modelling network data this means that all possible heterogeneity in the network nodes (that is the actors in the network) is included as covariates in the model and influence the (global) structure of the network. Since homogeneity of the nodes have led from p_1 to p_2 models, we want to pursue the same modelling exercise by allowing for latent node specific heterogeneity

in Exponential Random Graph Models. To do so, we combine the p_2 model (3.2) with the ERGM (3.4) towards

$$\text{logit}\left[\mathbb{P}\left(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\theta}, \phi_i, \phi_j\right)\right] = \boldsymbol{\theta}^t s_{ij}(\mathbf{y}) + \phi_i + \phi_j \quad (3.5)$$

with $\boldsymbol{\phi} = (\phi_1, \dots, \phi_n)^t$ and $\phi_i \stackrel{\text{i.i.d.}}{\sim} N(\mu_\phi, \sigma_\phi^2)$, $i = 1, \dots, n$. The parameter μ_ϕ captures the average propensity in the network for forming a tie. Therefore θ_0 , which is usually the parameter associated with the edges statistic, is excluded from $\boldsymbol{\theta}$, i.e. $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^t$ here. In terms of the likelihood function for the whole network we obtain from (3.5)

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \boldsymbol{\theta}, \boldsymbol{\phi}) = f(\mathbf{y} | \boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{q_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{y})}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})} = \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y}) + \boldsymbol{\phi}^t t(\mathbf{y})\}}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})}, \quad (3.6)$$

where $t(\mathbf{y})$ contains the degree statistics of the n vertices, i.e. $t_i(\mathbf{y}) = \sum_{j=1}^n y_{ij}$, for $i = 1, \dots, n$. That is we fit an Exponential Random Graph Model with random, node specific effects accounting for heterogeneity. The model in equations (3.5) and (3.6) falls in the general class of Exponential-family Random Network Models proposed by Fellows and Handcock (2012) but unlike their model we treat the node specific effect as latent and we pursue a fully Bayesian estimation. We also refer to Krivitsky et al. (2009) who propose a model with actor specific random effects based on a latent cluster model. The authors also propose node specific random effects. We follow this line and give further interpretability of the effects. A central issue in model extensions is the question of model selection. We emphasize this point in this work by comparing models with and without nodal effects using the Bayes factor as model selection criterion. However, calculation of the Bayes factor suffers from the above mentioned problem in Exponential Random Graph Models in that the normalization constant $\kappa(\cdot)$ is numerically infeasible. We therefore propose an approximate calculation of the Bayes factor and show in a simulation study its usability for model selection.

For estimation and model selection of model (3.6) we extend the fully Bayesian approach from Caimo and Friel (2011). The developed estimation routine is based on the numerical work of Caimo and Friel (2014) with their R (R Core Team, 2016) package **Bergm** (see <http://cran.r-project.org/web/packages/Bergm>). Our algorithms for model fitting and selection will be included in the **Bergm** package.

The chapter is organized as follows. In Section 3.2 we derive a fully Bayesian formulation of the model. This is followed by a detailed description of the MCMC based estimation routine. Section 3.3 deals with the issue of model selection using Bayes factors. Three data examples and some simulation results are presented in Section 3.4. Finally Section 3.5 concludes with a discussion.

3.2 Bayesian Model Formulation and Estimation

Before proposing a fully Bayesian formulation for model (3.6) bear in mind that the normalizing constant $\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})$ is numerically infeasible to calculate except for small networks so that numerically demanding simulation based fitting routines need to be employed. We follow a fully Bayesian approach by imposing a prior distribution on $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^t$. The posterior of interest for the Bayesian Exponential Random Graph Model with nodal random effects in (3.6) then becomes

$$p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mu_\phi, \sigma_\phi^2 | \mathbf{y}) = \frac{f(\mathbf{y} | \boldsymbol{\theta}, \boldsymbol{\phi}) p(\boldsymbol{\theta}) p(\boldsymbol{\phi} | \mu_\phi, \sigma_\phi^2) p(\mu_\phi) p(\sigma_\phi^2)}{p(\mathbf{y})}, \quad (3.7)$$

where $p(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$ and $p(\boldsymbol{\phi} | \mu_\phi, \sigma_\phi^2)$ the prior for the random nodal effects $\boldsymbol{\phi}$. We assume the nodal effects to be independent and identically normally distributed, that is

$$\phi_i \sim N(\mu_\phi, \sigma_\phi^2), \quad \text{for } i = 1, \dots, n$$

and accordingly we use $\boldsymbol{\theta} \sim N(0, \rho^2 I_p)$, with I_p denoting the p -dimensional unity matrix and ρ^2 chosen such that the prior distribution is flat. For the hyper prior distribution $p(\mu_\phi)$ of the mean μ_ϕ we assume a normal distribution centred at 0, that is

$$\mu_\phi \sim N(0, \tau^2).$$

The hyper prior $p(\sigma_\phi^2)$ of the variance σ_ϕ^2 is assumed to be an inverse gamma distribution, that is

$$\sigma_\phi^2 \sim IG(a, b).$$

Finally, the parameters τ^2 , a and b are all constants and chosen in a way that results in flat hyper prior distributions. Figure 3.1 illustrates this Bayesian model formulation.

It is important to note, that the posterior distribution in (3.7) is so-called doubly-intractable. This is because, firstly, it is not possible to evaluate the posterior density (3.7) due to $p(\mathbf{y})$, the marginal likelihood or evidence, being intractable. Secondly, it is also numerically infeasible to calculate the normalizing constant $\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})$ in the likelihood $f(\mathbf{y} | \boldsymbol{\theta}, \boldsymbol{\phi})$ except for very small network graphs. Similar to the algorithm proposed by Caimo and Friel (2011) we use the so-called exchange algorithm from Murray et al. (2006) to draw samples from the posterior distribution of interest. Let therefore $\boldsymbol{\gamma} = (\boldsymbol{\theta}^t, \boldsymbol{\phi}^t)^t$ denote the entire parameter vector of the ERGM. Instead of drawing directly from (3.7),

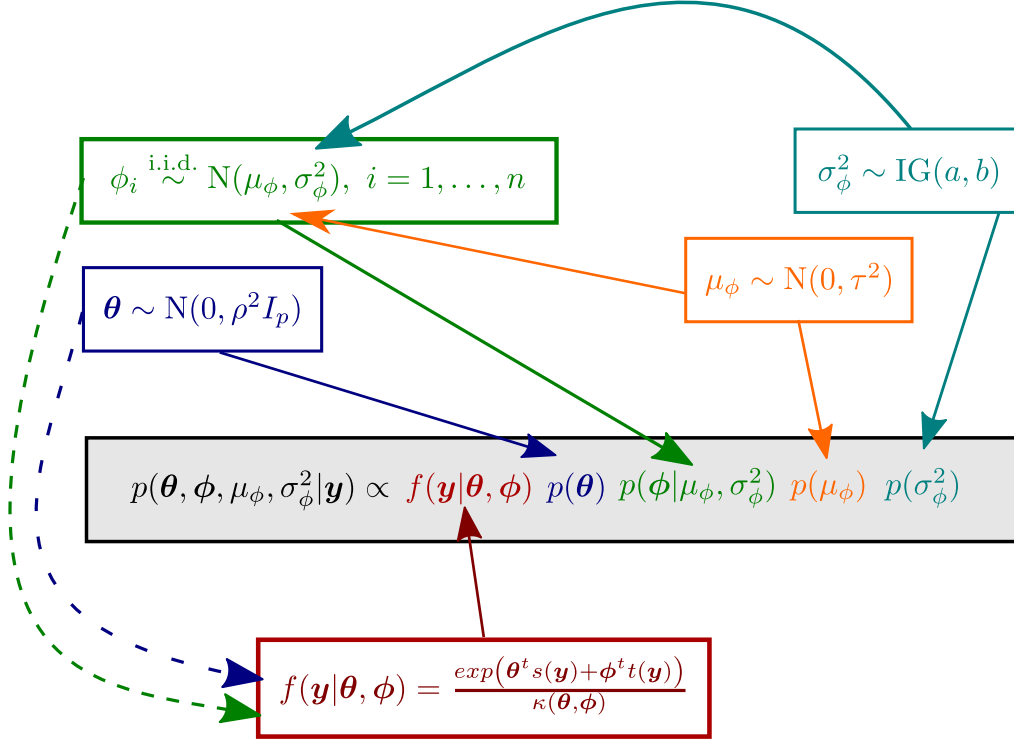


Figure 3.1 Overview of the Bayesian model formulation for the Exponential Random Graph Model with nodal random effects.

we sample from the augmented distribution

$$p(\gamma', \mathbf{y}', \gamma, \mu_\phi, \sigma_\phi^2 | \mathbf{y}) \propto f(\mathbf{y} | \gamma) p(\gamma | \mu_\phi, \sigma_\phi^2) p(\mu_\phi) p(\sigma_\phi^2) h(\gamma' | \gamma) f(\mathbf{y}' | \gamma'), \quad (3.8)$$

where $h(\cdot | \cdot)$ is a proposal function, to be specified later. This proposal provides $\gamma' = (\boldsymbol{\theta}^t, \boldsymbol{\phi}^t)^t$ as new candidate values for $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively, and based on γ' we can simulate \mathbf{y}' as an auxiliary network. The proposal is accepted with probability

$$\alpha = \min \left(1, \frac{q_\gamma(\mathbf{y}') p(\gamma') h(\gamma | \gamma') q_{\gamma'}(\mathbf{y})}{q_\gamma(\mathbf{y}) p(\gamma) h(\gamma' | \gamma) q_{\gamma'}(\mathbf{y}')} \times \frac{\kappa(\gamma) \kappa(\gamma')}{\kappa(\gamma') \kappa(\gamma)} \right), \quad (3.9)$$

where $p(\gamma) = p(\boldsymbol{\theta}) \cdot p(\boldsymbol{\phi} | \mu_\phi, \sigma_\phi^2)$. Note that in (3.9) the normalizing constants cancel out so that (3.9) is in principle easy to calculate. Though the algorithm is in this form a direct extension of the algorithm for Bayesian Exponential Random Graph Models of Caimo and

Friel (2011) it is advisable to separate the proposals of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ to achieve higher acceptance rates. This is described in the following algorithmic steps. In detail, our algorithm works as follows:

Algorithm 1: Fit BERGM with nodal random effects

Step 1: Gibbs update of $(\boldsymbol{\theta}', \mathbf{y}')$:

- (i) Draw $\boldsymbol{\theta}' \sim h(\cdot | \boldsymbol{\theta})$.
- (ii) Draw $\mathbf{y}' \sim p(\cdot | \boldsymbol{\theta}', \boldsymbol{\phi})$.
- (iii) Propose to move from $\boldsymbol{\theta}$ to $\boldsymbol{\theta}'$ with probability

$$\alpha = \min \left(1, \frac{q_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{y}') p(\boldsymbol{\theta}') h(\boldsymbol{\theta} | \boldsymbol{\theta}') q_{\boldsymbol{\theta}', \boldsymbol{\phi}}(\mathbf{y})}{q_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{y}) p(\boldsymbol{\theta}) h(\boldsymbol{\theta}' | \boldsymbol{\theta}) q_{\boldsymbol{\theta}', \boldsymbol{\phi}}(\mathbf{y}')} \times \frac{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi}) \kappa(\boldsymbol{\theta}', \boldsymbol{\phi})}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi}) \kappa(\boldsymbol{\theta}', \boldsymbol{\phi})} \right).$$

Step 2: Gibbs update of $(\boldsymbol{\phi}', \mathbf{y}')$:

- (i) Draw $\boldsymbol{\phi}' \sim g(\cdot | \boldsymbol{\phi})$.
- (ii) Draw $\mathbf{y}' \sim p(\cdot | \boldsymbol{\theta}, \boldsymbol{\phi}')$.
- (iii) Propose to move from $\boldsymbol{\phi}$ to $\boldsymbol{\phi}'$ with probability

$$\alpha = \min \left(1, \frac{q_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{y}') p(\boldsymbol{\phi}' | \mu_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}}^2) g(\boldsymbol{\phi} | \boldsymbol{\phi}') q_{\boldsymbol{\theta}, \boldsymbol{\phi}'}(\mathbf{y})}{q_{\boldsymbol{\theta}, \boldsymbol{\phi}}(\mathbf{y}) p(\boldsymbol{\phi} | \mu_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}}^2) g(\boldsymbol{\phi}' | \boldsymbol{\phi}) q_{\boldsymbol{\theta}, \boldsymbol{\phi}'}(\mathbf{y}')} \times \frac{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi}) \kappa(\boldsymbol{\theta}, \boldsymbol{\phi}')}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi}) \kappa(\boldsymbol{\theta}, \boldsymbol{\phi}')} \right).$$

Step 3: Metropolis-Hastings update of $\mu_{\boldsymbol{\phi}}$:

Draw proposal $\mu'_{\boldsymbol{\phi}}$ from $k(\cdot | \mu_{\boldsymbol{\phi}})$ and accept the proposed value with probability

$$\alpha = \min \left(1, \frac{p(\boldsymbol{\phi} | \mu_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}}^2) p(\mu_{\boldsymbol{\phi}})}{p(\boldsymbol{\phi} | \mu'_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}}^2) p(\mu'_{\boldsymbol{\phi}})} \right).$$

Step 4: Metropolis-Hastings update of $\sigma_{\boldsymbol{\phi}}^2$:

Draw proposal $\sigma_{\boldsymbol{\phi}}^{2'}$ from $l(\cdot | \sigma_{\boldsymbol{\phi}}^2)$ and accept the proposed value with probability

$$\alpha = \min \left(1, \frac{p(\boldsymbol{\phi} | \mu_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}}^2) p(\sigma_{\boldsymbol{\phi}}^2)}{p(\boldsymbol{\phi} | \mu_{\boldsymbol{\phi}}, \sigma_{\boldsymbol{\phi}}^{2'}) p(\sigma_{\boldsymbol{\phi}}^{2'})} \right).$$

Start again with *Step 1* until the maximum number of iterations is reached.

It is easy to see that there is no necessity to compute the normalizing constants $\kappa(\cdot)$, because they cancel out when calculating the acceptance probabilities in the first two steps of the algorithm. The current implementation of the algorithm uses single-site updates for the update of ϕ , that is each ϕ_i , $i = 1, \dots, n$ is updated in turn while all other values are kept constant. This leads to reasonable acceptance probabilities for the Markov chain.

The default choices for the proposal functions $h(\cdot|\cdot)$, $g(\cdot|\cdot)$ and $k(\cdot|\cdot)$ are normal distributions centred at the current parameter value, for $l(\cdot|\cdot)$ we use a uniform distribution, which is symmetric around the current value of σ_ϕ^2 and truncated at zero to avoid negative proposals for the variance parameter.

The draws of the auxiliary network \mathbf{y}' in the component of steps 1 and 3 are realised using the “tie no tie” sampler from the `ergm` package (Hunter et al., 2008b), which is a simple Gibbs sampler. Although this auxiliary Gibbs sampler does not yield an exact draw \mathbf{y}' , Everitt (2012) has shown, under some assumptions, that the resulting approximate exchange algorithm converges to the target distribution as the number of auxiliary draws tends to infinity. As a practical result he points out that for the number of auxiliary iterations it is often sufficient to use roughly the number of possible ties in the network.

3.3 Model Selection

3.3.1 Bayesian Model Selection

Model Selection is an important, often neglected issue in network data analysis. We put special emphasis on this task here and propose the Bayes factor suitable for model selection. One of the interesting questions in our model is, if we are able to distinguish the three following model generating processes:

- (1) Nodal random effects only, i.e. the p_2 model,
- (2) Structural effects only, i.e. the standard ERGM, and
- (3) ERGM in combination with nodal random effects.

This question results in the problem of model selection. The data examples in Section 3.4.1 illustrate this issue.

Classical Bayesian tools for model comparison such as the deviance information criterion (DIC) as suggested by Spiegelhalter et al. (2002) are not directly available, again due to the intractability of the normalizing constant of the likelihood in model equation (3.6).

Computing Bayes factors for model choice using reversible jump Markov Chain Monte Carlo for Bayesian Exponential Random Graph Models as done by Caimo and Friel (2013) is not an option for our model. This approach would be possible in general, but very time consuming from a computational point of view.

In the following subsections we present two approaches for model selection based on Bayes factors, one for nested models, and a more general approach for non-nested models.

3.3.2 Bayes Factor for Nested Models

We suggest the following strategy for deciding whether to include nodal random effects into the model or not. The goal is to calculate a Bayes factor for two competing models (Kass and Raftery, 1995). First we fit the two Exponential Random Graph Models, one with edges and non-random effects only, notated as model m_1 with coefficients $\boldsymbol{\theta}' = (\theta'_0, \dots, \theta'_p)^t$, and the second one with nodal random effects instead of the edges term, labelled as model m_2 with coefficients $\boldsymbol{\theta} = (\theta_1, \dots, \theta_p)^t$ and $\boldsymbol{\phi} = (\phi_1, \dots, \phi_n)^t$. For the moment we assume that the two models are nested, that is the statistics $s_1(\mathbf{y}), \dots, s_p(\mathbf{y})$ are the same in models m_1 and m_2 . In the next subsection we present a more general approach for non-nested models. Following Bayes theorem the so-called evidence for each model can be calculated using

$$p(\mathbf{y}|m_1) = \frac{f(\mathbf{y}|\boldsymbol{\theta}')p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta}'|\mathbf{y})}, \quad \forall \boldsymbol{\theta}', \quad (3.10)$$

for model m_1 , and

$$\begin{aligned} p(\mathbf{y}|m_2) &= \frac{f(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\phi}, \mu_\phi, \sigma_\phi^2)p(\boldsymbol{\theta})p(\boldsymbol{\phi}|\mu_\phi, \sigma_\phi^2)p(\mu_\phi)p(\sigma_\phi^2)}{p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mu_\phi, \sigma_\phi^2|\mathbf{y})}, \quad \forall \boldsymbol{\theta}, \boldsymbol{\phi}, \mu_\phi, \sigma_\phi^2, \\ &= \frac{f(\mathbf{y}|\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2)p(\boldsymbol{\theta})p(\mu_\phi)p(\sigma_\phi^2)}{p(\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2|\mathbf{y})}, \quad \forall \boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2, \end{aligned} \quad (3.11)$$

for model m_2 .

The term $f(\mathbf{y}|\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2)$ denotes the marginal likelihood from model m_2 , where the random effects $\boldsymbol{\phi}$ have been marginalized, i.e.

$$\begin{aligned} f(\mathbf{y}|\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2) &= \int \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y}) + \boldsymbol{\phi}^t t(\mathbf{y})\}}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})} \cdot p(\boldsymbol{\phi}|\mu_\phi, \sigma_\phi^2) d\boldsymbol{\phi} \\ &\approx \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}}{\kappa(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})} \hat{f}_{\text{Laplace}}(\mathbf{y}|\hat{\boldsymbol{\phi}}, \mu_\phi, \sigma_\phi^2). \end{aligned} \quad (3.12)$$

The approximation in equation (3.12) is achieved using a Laplace approximation around the point $\hat{\boldsymbol{\phi}}$. Details of this approximation are given in Section C of the appendix.

The Bayes factor of model m_2 against model m_1 is then defined as the ratio of (3.11) and

(3.10), i.e.

$$\text{BF}_{21} = \frac{p(\mathbf{y}|m_2)}{p(\mathbf{y}|m_1)} = \frac{f(\mathbf{y}|\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2)}{f(\mathbf{y}|\boldsymbol{\theta}')} \cdot \frac{p(\boldsymbol{\theta})p(\mu_\phi)p(\sigma_\phi^2)}{p(\boldsymbol{\theta}')} \cdot \frac{p(\boldsymbol{\theta}'|\mathbf{y})}{p(\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2|\mathbf{y})}. \quad (3.13)$$

Applying the approximation from equation (3.12) to (3.13), and plugging in estimates for the posterior densities

$$p(\boldsymbol{\theta}'|\mathbf{y}) \approx \hat{p}(\boldsymbol{\theta}'|\mathbf{y}) \quad \text{and} \quad p(\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2|\mathbf{y}) \approx \hat{p}(\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2|\mathbf{y}) \quad (3.14)$$

leads to

$$\text{BF}_{21} \approx \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y})\} \hat{f}_{\text{Laplace}}(\mathbf{y}|\hat{\boldsymbol{\phi}}, \mu_\phi, \sigma_\phi^2)}{\exp\{\boldsymbol{\theta}'^t s'(\mathbf{y})\}} \cdot \frac{\kappa(\boldsymbol{\theta}')}{\kappa(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})} \cdot \frac{p(\boldsymbol{\theta})p(\mu_\phi)p(\sigma_\phi^2)}{p(\boldsymbol{\theta}')} \cdot \frac{\hat{p}(\boldsymbol{\theta}'|\mathbf{y})}{\hat{p}(\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2|\mathbf{y})}. \quad (3.15)$$

The ratio of the two normalizing constants $\kappa(\boldsymbol{\theta}') / \kappa(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})$ in (3.15) is estimated using a path sampling approach (Gelman and Meng, 1998), which is similarly used by Caimo and Friel (2013). Consider

$$\kappa(\boldsymbol{\theta}(g), \boldsymbol{\phi}(g)),$$

where

$$\boldsymbol{\theta}(g) = (1 - g) \cdot \boldsymbol{\theta}' + g \cdot \begin{bmatrix} 0 \\ \boldsymbol{\theta} \end{bmatrix} \quad \text{and}$$

$$\boldsymbol{\phi}(g) = g \cdot \boldsymbol{\phi}$$

for $g \in [0, 1]$. So by construction

$$(\boldsymbol{\theta}(0), \boldsymbol{\phi}(0)) = (\boldsymbol{\theta}', \mathbf{0}) \quad \text{and} \quad (\boldsymbol{\theta}(1), \boldsymbol{\phi}(1)) = \left(\begin{bmatrix} 0 \\ \boldsymbol{\theta} \end{bmatrix}, \boldsymbol{\phi} \right).^1$$

Then thermodynamic integration (or so-called path sampling) can be used to estimate

$$\log \left\{ \frac{\kappa(\boldsymbol{\theta}')}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})} \right\} = \int_0^1 \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}(g), \boldsymbol{\phi}(g)} \left[\left(\boldsymbol{\theta}' - \begin{bmatrix} 0 \\ \boldsymbol{\theta} \end{bmatrix} \right)^t s'(\mathbf{Y}) + (-\boldsymbol{\phi})^t t(\mathbf{Y}) \right] dg.$$

¹ Note that the additional 0 entry is necessary because the mixed effects model contains no parameter for the edges statistic. The edge effect is captured in the mean value μ_ϕ of the nodal random effects. The missing edges statistics is also the difference between $s'(\mathbf{y})$ and $s(\mathbf{y})$.

Consider discretising $g \in [0, 1]$ as $(g_0 = 0, \dots, g_i = \frac{i}{I}, \dots, g_I = 1)$. Then we approximate

$$\begin{aligned} E_i &:= \mathbb{E}_{\mathbf{Y}|\boldsymbol{\theta}(g_i), \boldsymbol{\phi}(g_i)} \left[\left(\boldsymbol{\theta}' - \begin{bmatrix} 0 \\ \boldsymbol{\theta} \end{bmatrix} \right)^t s'(\mathbf{Y}) + (-\boldsymbol{\phi})^t t(\mathbf{Y}) \right] \\ &\approx \frac{1}{N} \sum_{j=1}^N \left[\left(\boldsymbol{\theta}' - \begin{bmatrix} 0 \\ \boldsymbol{\theta} \end{bmatrix} \right)^t s'(\mathbf{y}^{(j)}) + (-\boldsymbol{\phi})^t t(\mathbf{y}^{(j)}) \right], \end{aligned}$$

where the networks $\mathbf{y}^{(j)}$ are drawn from $f(\mathbf{y}|\boldsymbol{\theta}(g_i), \boldsymbol{\phi}(g_i))$, for $j = 1, \dots, N$. Then we use a trapezoidal rule to numerically integrate

$$\log \left\{ \frac{\kappa(\boldsymbol{\theta}')}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})} \right\} = \sum_{i=1}^{I-1} (g_{i+1} - g_i) \cdot \left(\frac{E_{i+1} + E_i}{2} \right).$$

The path sampling routine can easily be parallelised because the evaluations at the individual grid points of g do not depend on each other.

The Bayes factor in equation (3.15) is evaluated using the posterior mean values for the parameters $\boldsymbol{\theta}$, $\boldsymbol{\theta}'$, μ_ϕ and also for $\hat{\boldsymbol{\phi}}$. For σ_ϕ^2 we plug in the mean of the logarithmized values and transform it back onto the scale of σ_ϕ^2 , because the posterior density of σ_ϕ^2 is not symmetric.

For reasons of simplicity the posterior density estimates (3.14) are estimated assuming asymptotic normality, again using $\log(\sigma_\phi^2)$. For the data examples in the next section this assumption seems to be reasonable when looking at the plotted posterior density estimates. Furthermore, the individual contributions of the different components of the Bayes factor calculation suggest that at least in these cases the posterior density estimates play a minor part compared to the other components. If this assumption is violated this step in the algorithm can be changed.

3.3.3 Bayes Factor for Non-Nested Models

The extension of the Bayes factor to allow for a comparison of non-nested models is relatively straightforward. Let m_a and m_b be two not necessarily nested Bayesian Exponential Random Graph Models, with or without random effects. We denote the corresponding parameter vectors for each model with $\boldsymbol{\gamma}_a$ and $\boldsymbol{\gamma}_b$, respectively. The Bayes factor is defined through

$$\text{BF}_{ba}^{\text{non-nested}} = \frac{p(\mathbf{y}|m_b)}{p(\mathbf{y}|m_a)} = \frac{f(\mathbf{y}|\boldsymbol{\gamma}_b)}{f(\mathbf{y}|\boldsymbol{\gamma}_a)} \cdot \frac{p(\boldsymbol{\gamma}_b)}{p(\boldsymbol{\gamma}_a)} \cdot \frac{p(\boldsymbol{\gamma}_a|\mathbf{y})}{p(\boldsymbol{\gamma}_b|\mathbf{y})}. \quad (3.16)$$

We distinguish three model setups. First (case 1), the model has fixed effects and nodal random effects, i.e. $\gamma = (\theta^t, \phi^t)^t$. Second (case 2), the model has random nodal effects only, i.e. a classical p_2 model (see van Duijn et al., 2004) with parameters $\gamma = \phi$. Third (case 3), the model is a regular Exponential Random Graph Model without nodal random effects, i.e. the parameters result to $\gamma = \theta$. If the model contains random effects we use a Laplace approximation for the corresponding likelihood component $f(\mathbf{y}|\gamma)$ as in equation (3.12). The components of the approximated Bayes factor in equation (3.15) concerning prior and estimated posterior densities remain the same as in equation (3.15), again depending on whether each model contains nodal random effects or not. The evidence for each model in (3.16) can be approximated by

$$p(\mathbf{y}|m) \approx \begin{cases} \frac{\exp\{\theta^t s(\mathbf{y})\} \hat{f}_{\text{Laplace}}(\mathbf{y}|\hat{\phi}, \mu_\phi, \sigma_\phi^2)}{\kappa(\theta, \hat{\phi})} \cdot \frac{p(\theta)p(\mu_\phi)p(\sigma_\phi^2)}{\hat{p}(\theta, \mu_\phi, \sigma_\phi^2|\mathbf{y})}, & \text{(case 1),} \\ \frac{\hat{f}_{\text{Laplace}}(\mathbf{y}|\hat{\phi}, \mu_\phi, \sigma_\phi^2)}{\kappa(\hat{\phi})} \cdot \frac{p(\mu_\phi)p(\sigma_\phi^2)}{\hat{p}(\mu_\phi, \sigma_\phi^2|\mathbf{y})}, & \text{(case 2),} \\ \frac{\exp\{\theta^t s(\mathbf{y})\}}{\kappa(\theta)} \cdot \frac{p(\theta)}{\hat{p}(\theta|\mathbf{y})}, & \text{(case 3).} \end{cases}$$

The major change concerns the estimation of the ratio of the two normalizing constants in (3.15) via path sampling. Instead of estimating the ratio directly we estimate two ratios,

$$\kappa(\mathbf{0}) / \kappa(\gamma_a) \quad \text{and} \quad \kappa(\mathbf{0}) / \kappa(\gamma_b),$$

via path sampling, where $\kappa(\mathbf{0})$ is the normalizing constant of a null model where all parameters are set to 0. The path sampling works in the same way as described before, but we substitute one of the models by the null model. By dividing the two estimated ratios, the term $\kappa(\mathbf{0})$ cancels out and we obtain an estimate of $\kappa(\gamma_a) / \kappa(\gamma_b)$. It should be clear that this approach takes almost double the amount of time (or cores) to estimate the ratio of normalizing constants as in the nested case where we only need one instance of path sampling. This suggests to use the Bayes factor (3.15) if the models are nested.

3.4 Examples

3.4.1 Data Examples

Zachary's Karate Club Network

As a first data example we employ Zachary's karate club network (Zachary, 1977) which is a very well known data set often used in network analysis. The undirected 34 node network represents the friendships among members of a university karate club. Figure 3.2 shows a plot of this network graph. It is evident that there are only some nodes with a very high degree (no. 1, 33, and 34) while the majority of the remaining vertices has only two to four links. If there are no additional nodal attributes available, that might explain some differences between the actors, like for example status in the club (trainer, student, etc.), the assumption of vertex homogeneity in a standard ERGM appears to be at least questionable.

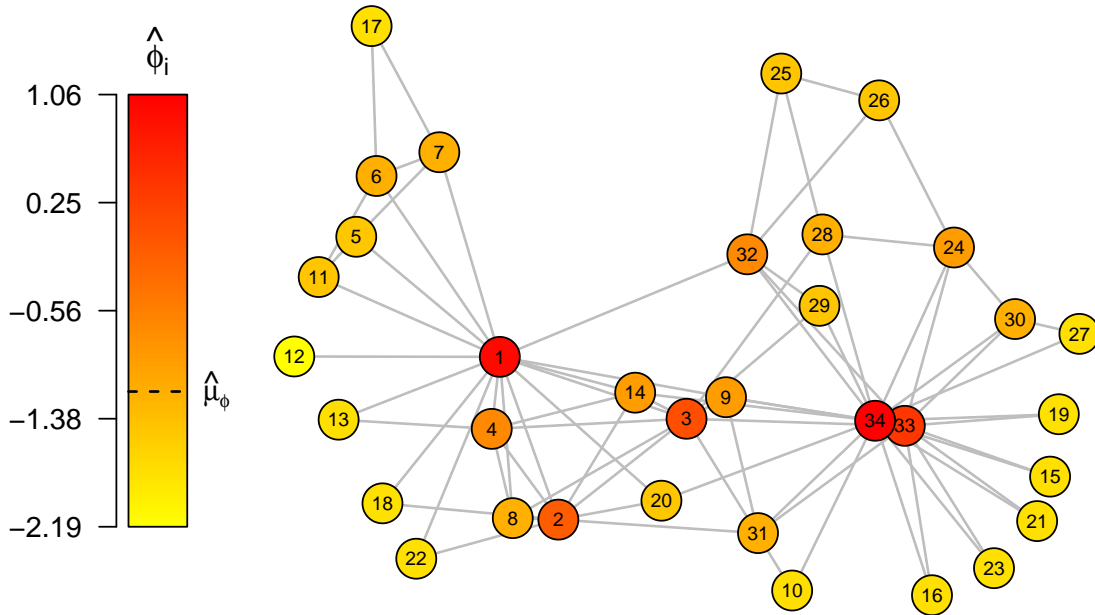


Figure 3.2 Zachary's Karate Club Graph. Vertices are coloured by their estimated nodal effect $\hat{\phi}_i$ (posterior mean), $i = 1, \dots, n$. Vertices with a high nodal effect are darker in orange/red.

As a first step, we fitted two different models to the data: a standard ERGM with edges and triangles as sufficient statistics, and a model with nodal random effects and the triangle statistic. These two models are nested.

For the model fitting tasks we used the **Bergm** package (Caimo and Friel, 2014) and our extension of the **Bergm** routines, respectively. With 1,000 burn-in iterations, 30,000 main iterations, and 3,000 auxiliary iterations for the network simulation in each MCMC step, the computation of the fixed model took about two minutes on a 2.1 Ghz processor, the mixed model needed about one hour and forty minutes. Using 3,000 auxiliary iterations should be large enough because we have 561 possible ties in the network. Again, we refer to the results of Everitt (2012).

Figure 3.3 shows the results for the fixed model with edges and triangular effect only. Figure 3.4 shows the results for the mixed model with nodal random and triangular effects for the karate club data. Table 3.1 shows the resulting posterior estimates for both models.

The vertices of the karate network in Figure 3.2 are coloured according to their estimated nodal effect $\hat{\phi}_i$, $i = 1, \dots, n$. As an estimate we use the corresponding posterior mean of each parameter ϕ_i . Darker coloured vertices (orange/red) correspond to those with a high nodal value. By using such a colouring scheme we are able to visualise the variation in the nodal effects. In addition, we can identify important nodes in the network based on the estimated nodal effects.

Figure 3.5 shows estimates for the posterior densities for both models simultaneously to allow for a visual comparison. What is evident from the estimated posterior densities is the difference for the triangular effect in both models in the upper right plot of Figure 3.5. When not accounting for nodal heterogeneity this effect is clearly positive compared to the

Table 3.1 *Model fitting results for the karate club data. The fixed model contains edges and triangles, and the mixed model triangles and nodal random effects.*

Model type	Parameter	Post. mean	Post. Sd.	Acceptance rate	Note
fixed	θ_{edges}	-2.32	0.16	0.43	
	$\theta_{\text{triangles}}$	0.54	0.11		
mixed	μ_{ϕ}	-1.17	0.22	0.26	*
	σ_{ϕ}^2	1.05	0.58	0.54	
	$\theta_{\text{triangles}}$	-0.04	0.21	0.09	

* For σ_{ϕ}^2 the posterior mean is calculated based on the logarithmized values and then transformed back to the scale of σ_{ϕ}^2 (this leads to the geometric mean) due to the non-symmetric posterior density in this case.

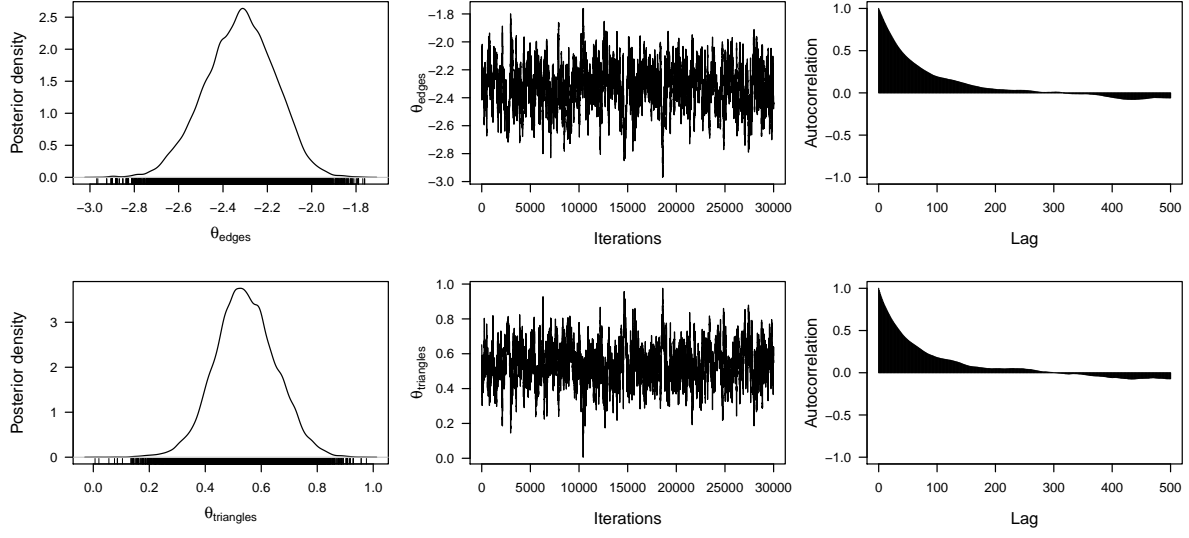


Figure 3.3 Posterior densities, trace plots, and autocorrelation for the fixed model with edges and triangular effect for the karate club data.

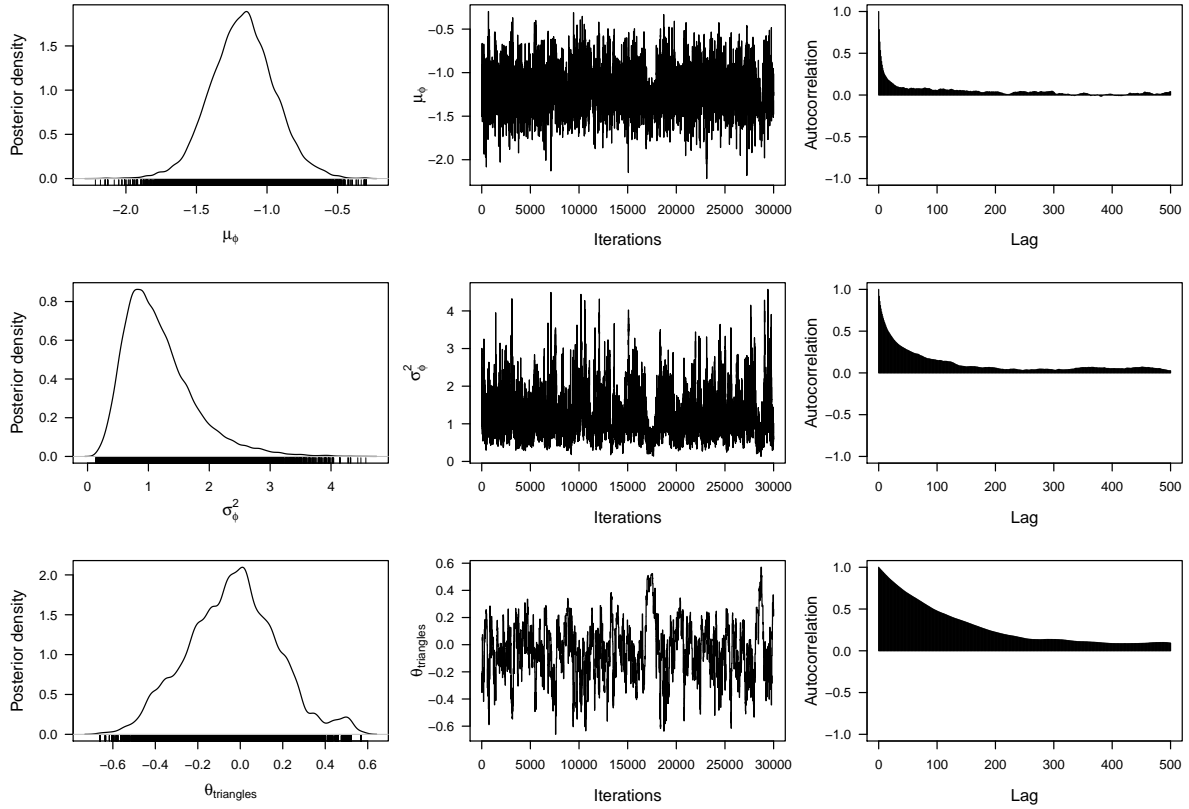


Figure 3.4 Posterior densities, trace plots, and autocorrelation for the mixed model with nodal random and triangular effects for the karate club data.

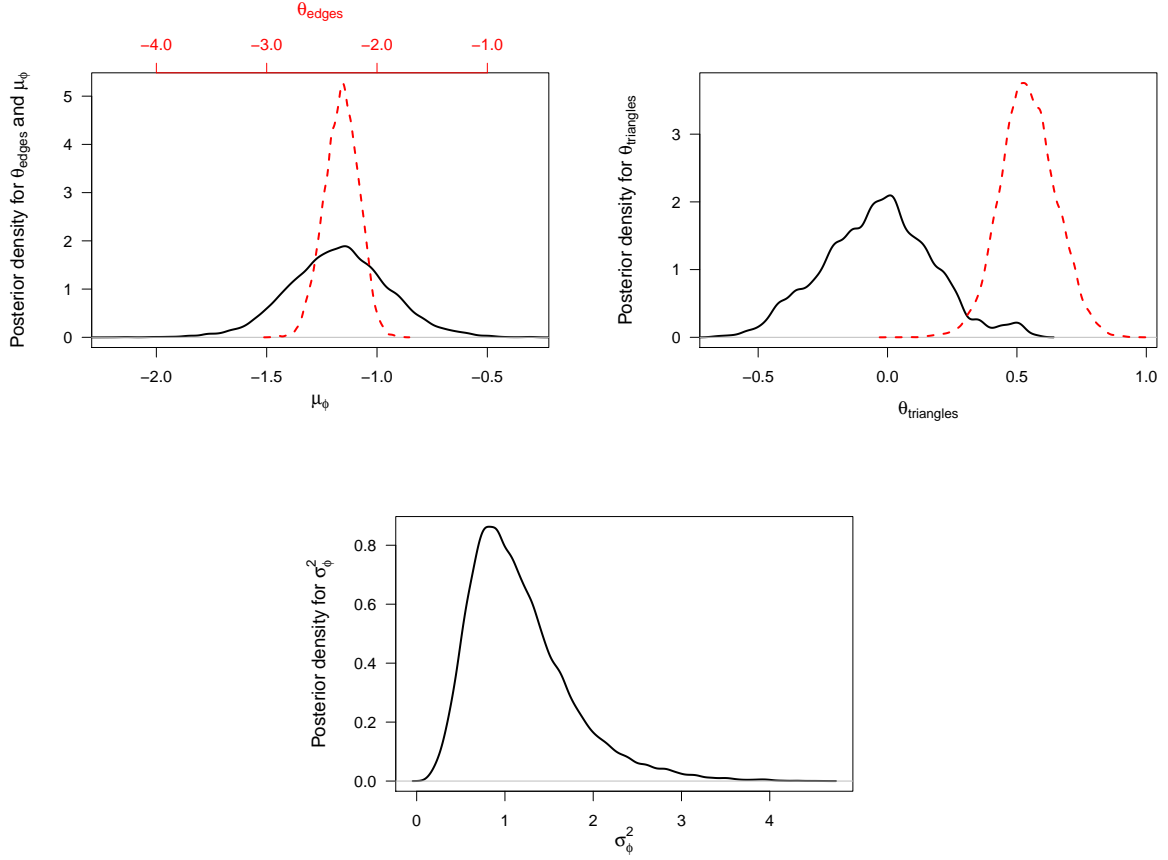


Figure 3.5 Posterior densities for the model with edges and triangular effect (red dashed lines) and nodal random and triangular effects (black solid lines) for the karate club data.

mixed model where the posterior support clearly comprises zero. For the parameter θ_{edges} associated with the edges statistic and μ_ϕ in the mixed model there is no big difference between both models concerning the location of the posterior (when comparing θ_{edges} to $2 \cdot \mu_\phi$; note the different axis annotations).

Figure 3.6 and Figure 3.7 show Bayesian goodness-of-fit plots for both models. For each model we used 100 draws from the corresponding posterior and simulated a network for each of the posterior parameter combinations. Boxplots of the distributions of degree, minimum geodesic distance, and edge-wise shared partners for the resulting simulated networks are shown in the plots where the bold red line indicates the values of the original karate club network. If other aspects of the data are of interest in order to assess goodness-of-fit of a model, e.g., triad census (see, e.g., Caimo and Friel, 2011, Figure 15) the goodness-of-fit plots can be customized accordingly. For the fixed model with edges and triangle effect we see some problems in Figure 3.6 concerning especially the degree distribution and the edge-wise shared partner distribution. For some (but not all) of the resulting simulated networks we have a high proportion of nodes with degree 33, which corresponds to a full or

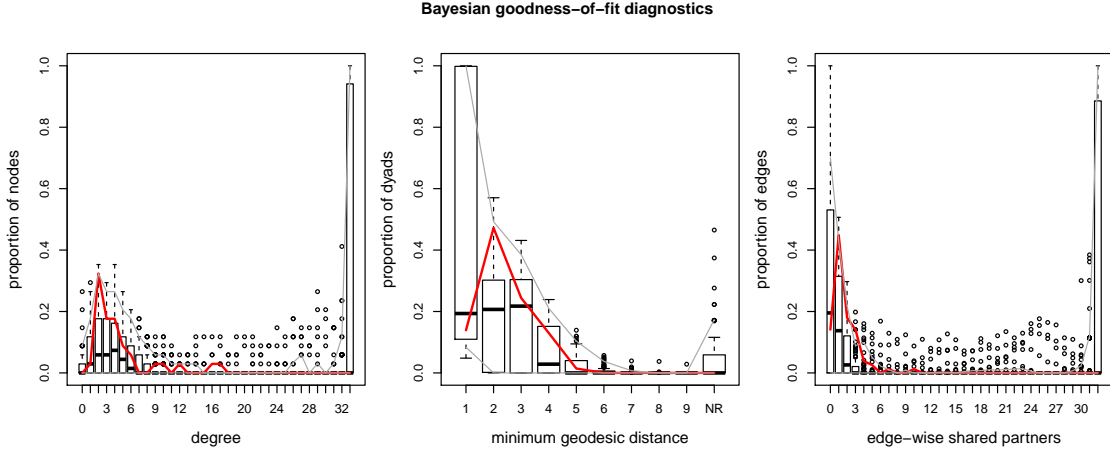


Figure 3.6 *Bayesian goodness-of-fit diagnostics for the fixed model with edges and triangle effect for the karate club data. Bold red line corresponds to original dataset.*

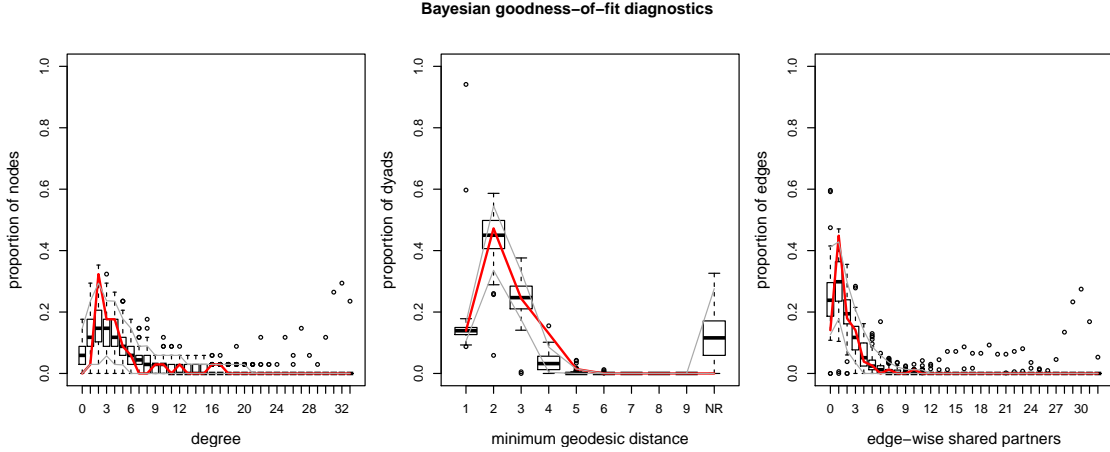


Figure 3.7 *Bayesian goodness-of-fit diagnostics for the mixed model with triangle and nodal random effects for the karate club data. Bold red line corresponds to original dataset.*

an almost full network. The same applies to very high proportions of edges with 32 edge-wise shared partners. It is well known from the literature, see, for example, Lusher et al. (2013), Chatterjee and Diaconis (2013), and Schweinberger (2011), that an Exponential Random Graph Model containing only edges and triangles as statistics is problematic, i.e. we have degeneracy issues. In the Bayesian setting here the model with edges and triangles does not result in complete degeneracy as only some but not all of the resulting simulated networks are complete graphs, which coincides with the findings of (Caimo and Friel, 2011). For the simulation of each network we use 10,000 iterations which is sufficiently large for a network with 34 nodes to assume convergence of the underlying chain. For the mixed model with triangles and nodal random effects the resulting plots in Figure 3.7 look reasonable. Especially the different effect of the triangular statistic in both models clearly illustrates the issue of model selection. After fitting the two competing models we computed a Bayes

factor using the approach for nested models described in Section 3.3.2 to compare the model with nodal random effects to the one with structural effects only and tackle this issue.

The resulting estimated log Bayes factor is 453, which is huge. As explained in the previous section, there is some randomness involved in the procedure. Repeated calculation led to similarly huge values. This clearly indicates that the model with nodal random effects is preferable to the one without and is not surprising, because here a model with nodal heterogeneity appears much more realistic than one without.

Computing a single Bayes factor took about fourteen minutes using five 2.2 GHz cores in parallel, with 10,000 iterations for the Laplace approximation, 1,000 grid points, 1,000 iterations at each point for the path sampling, and 3,000 iterations for each network simulation.

As second step of our analysis of the karate data we have used a model containing edges and the geometrically weighted edge-wise shared partner (GWESP) statistic

$$e^{\theta_{\text{dec}}} \sum_{i=1}^{n-2} \left\{ 1 - \left(1 - e^{-\theta_{\text{dec}}} \right)^i \right\} \text{EP}_i(\mathbf{y}),$$

where $\text{EP}_i(\mathbf{y})$ denotes the number of edges with i shared partners, see Hunter (2007) for details. Adding this term to the model circumvents the degeneracy issue known from the edges and triangle model. Even though it can be shown that the term is equivalent to

Table 3.2 *Model fitting results for the karate club data. The fixed model contains edges and geometrically weighted edge-wise shared partners (GWESP), and the mixed model GWESP and nodal random effects. For both models the decay parameter for GWESP is fixed at 0.8. The random model contains only the nodal random effects.*

Model type	Parameter	Post. mean	Post. Sd.	Acceptance rate	Note
fixed	θ_{edges}	-2.99	0.24	0.37	
	θ_{gwesp}	0.63	0.11		
mixed	μ_{ϕ}	-1.22	0.21	0.24	*
	σ_{ϕ}^2	0.88	0.47	0.48	
	θ_{gwesp}	0.08	0.13	0.06	
random	μ_{ϕ}	-1.18	0.20	0.25	*
	σ_{ϕ}^2	1.01	0.45	0.53	

* For σ_{ϕ}^2 the posterior mean is calculated based on the logarithmized values and then transformed back to the scale of σ_{ϕ}^2 (this leads to the geometric mean) due to the non-symmetric posterior density in this case.

the alternating k -triangle statistic, see Hunter (2007), and Snijders et al. (2006), from a modelling point of view, both terms, GWESP and the alternating k -triangles are more complicated to interpret. By setting the decay parameter to a fixed value, in our case $\theta_{\text{dec}} = 0.8$, the model is a regular, non-curved Exponential Random Graph Model (Hunter and Handcock, 2006).

In addition we have fitted a mixed model with nodal random effects and the GWESP statistic, again with a fixed decay parameter $\theta_{\text{dec}} = 0.8$, and a model containing only the nodal random effects, which is just the Bayesian version of a p_2 model (van Duijn et al., 2004). Table 3.2 shows the resulting posterior estimates for all three models.

Figure 3.8 shows the results for the fixed model with edges and geometrically weighted edge-wise shared partners effect. The resulting trace and autocorrelation plots for both parameters reveal that the Markov chain did not mix so well and this could have been alleviated by thinning the chain. Figure 3.9 shows the results for the mixed model with geometrically weighted edge-wise shared partners (with fixed decay of 0.8) and nodal random effects for the karate club data. The trace plot and autocorrelation plot for θ_{gwesp} convey that the Markov chain did not mix as well for this parameter compared to the other two parameters. Figure 3.10 shows the results for the model with random effects only for the karate club data. The posterior density for the parameter associated with the GWESP statistic is almost centred at zero in the mixed model, while in the fixed model the parameter is clearly positive. So we see a comparable behaviour as in the models before for the triangle effect which is positive in the fixed model and becomes zero when nodal random effects are included in the model.

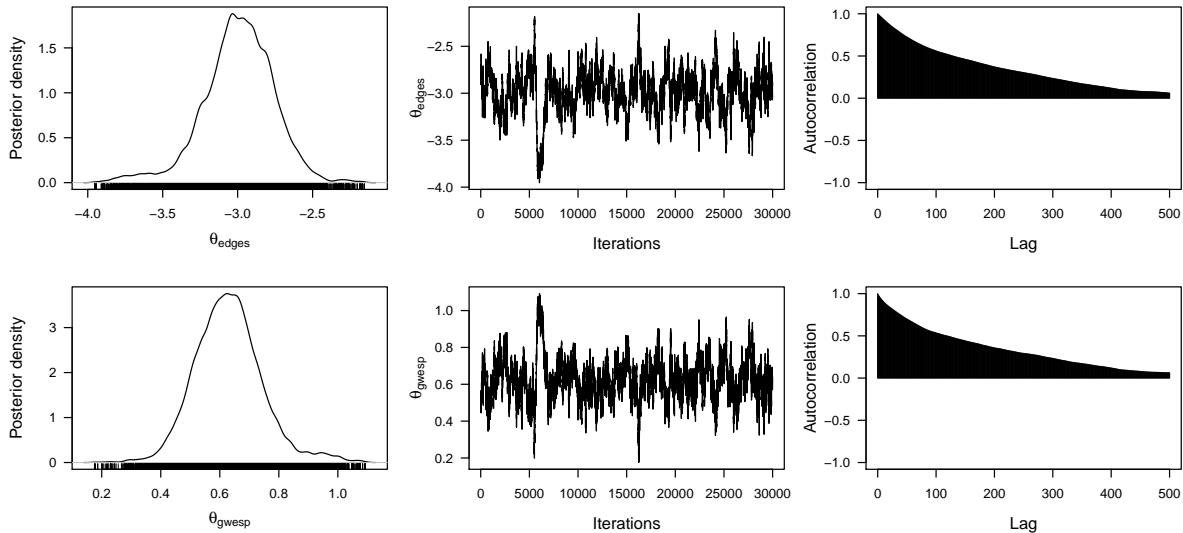


Figure 3.8 *Posterior densities, trace plots, and autocorrelation for the fixed model with edges and geometrically weighted edge-wise shared partners (with fixed decay of 0.8) effect for the karate club data.*

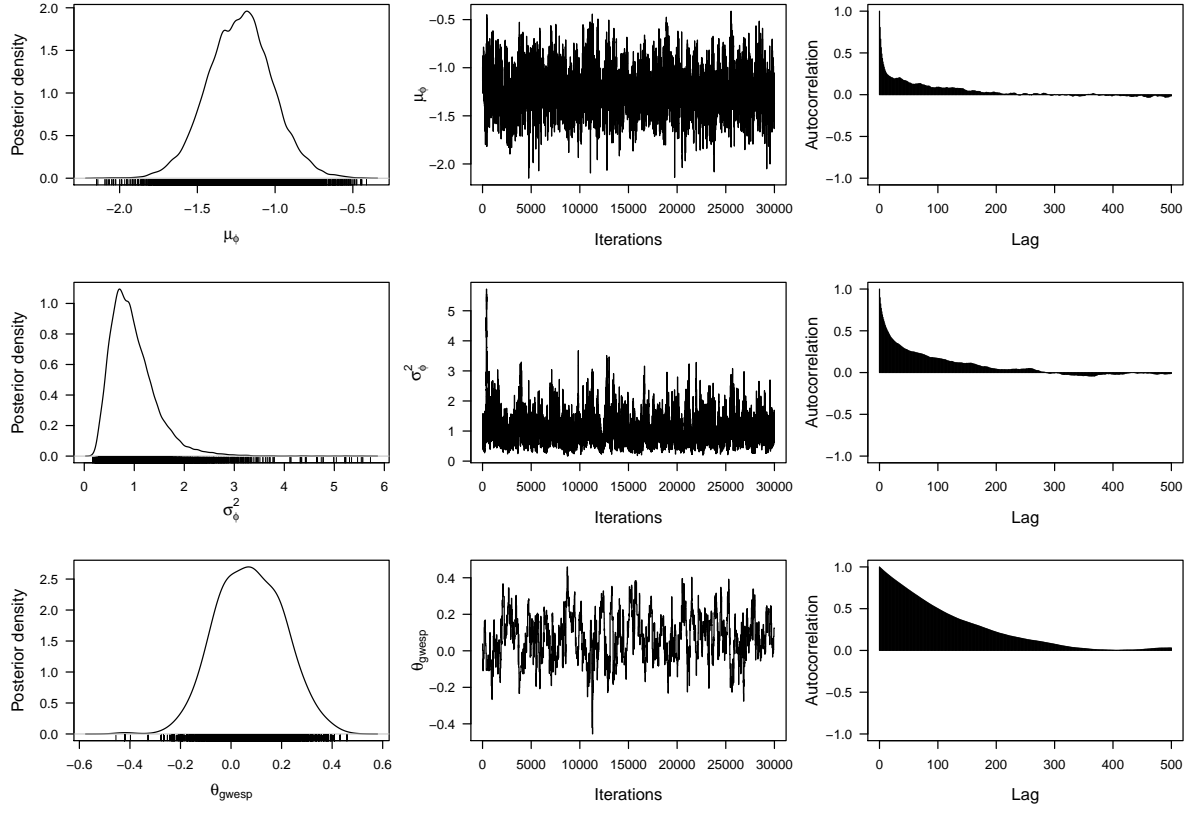


Figure 3.9 *Posterior densities, trace plots, and autocorrelation for the mixed model with geometrically weighted edge-wise shared partners (with fixed decay of 0.8) and nodal random effects for the karate club data.*

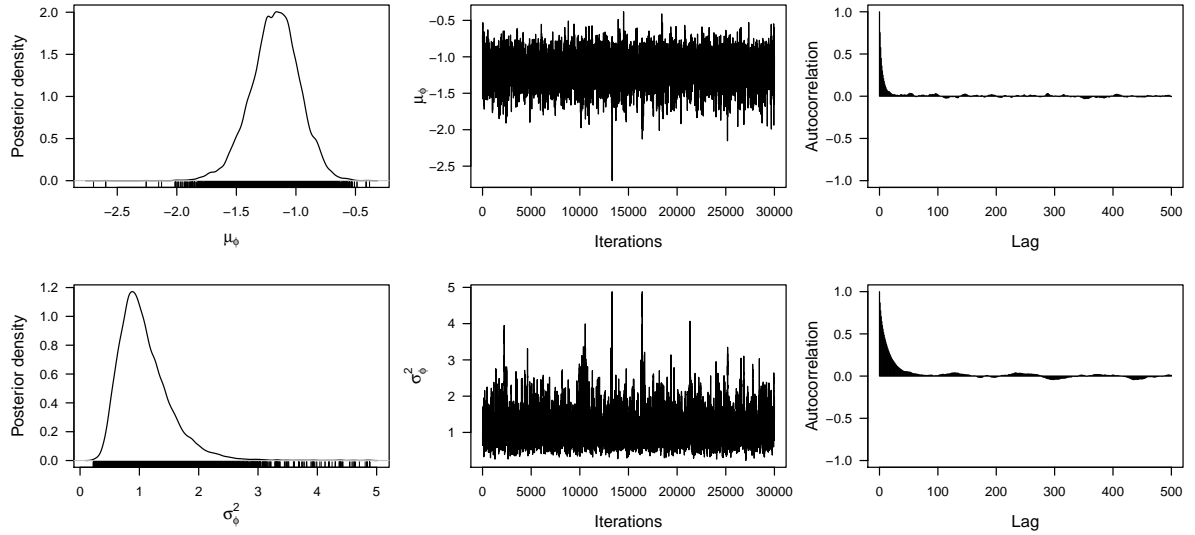


Figure 3.10 *Posterior densities, trace plots, and autocorrelation for the model with nodal random effects for the karate club data.*

Figures 3.11 – 3.13 show the corresponding goodness-of-fit diagnostic plots for the three models.² For the distribution of the minimum geodesic distance the model with edges and geometrically weighted edge-wise shared partner statistic seems to be better in capturing this property of the network than the two models with nodal random effects. In general, the resulting goodness-of-fit plots are comparable for all three models. Here we do not see any degenerate behaviour, i.e. there is no high proportion of nodes with degree 33 or a high proportion of edges with 32 shared partners which would be the case in a full graph.

For model selection in the second part of the karate data analysis we have computed two Bayes factors. The resulting log Bayes factor for the nested comparison of the fixed model with edges and geometrically weighted edge-wise shared partners against the mixed model with GWESP and nodal random effects is -0.41 . This value points into the direction of the fixed model, but is rather close to zero. So the conclusion here is that none of the two competing models is clearly better than the other. The second log Bayes factor for the non-nested comparison of the fixed model with edges and GWESP against the model with nodal random effects only is 3.83 , which is clearly in favour of the random effects model. So based on these values the model containing only nodal random effects would be preferred.

² The axis for the goodness-of-fit plots 3.11 – 3.13 has been changed to range from zero to one to achieve better comparability of the different plots. Figure 3.12, and the captions have been corrected compared to the original submission.

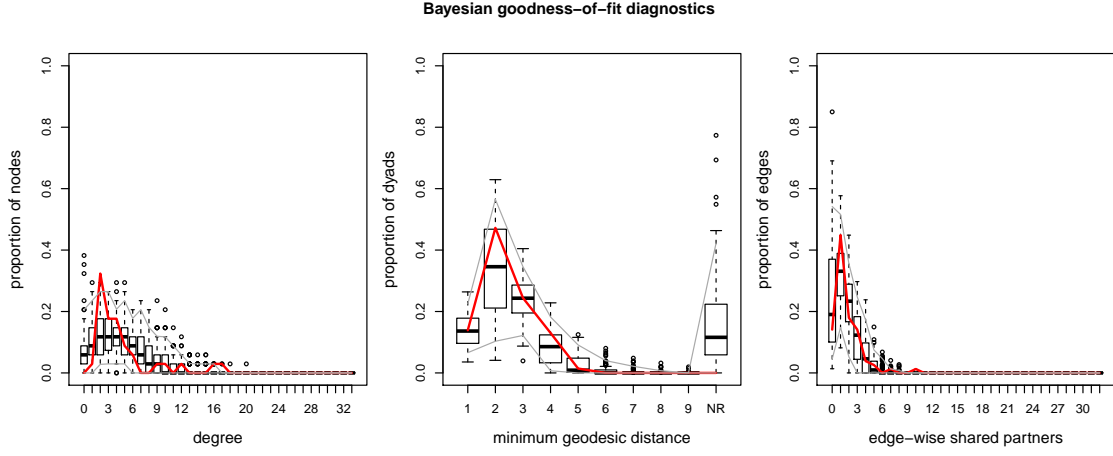


Figure 3.11 Bayesian goodness-of-fit diagnostics for the fixed model with edges and GWESP (with fixed decay of 0.8) for the karate club data. Bold red line corresponds to original dataset.²

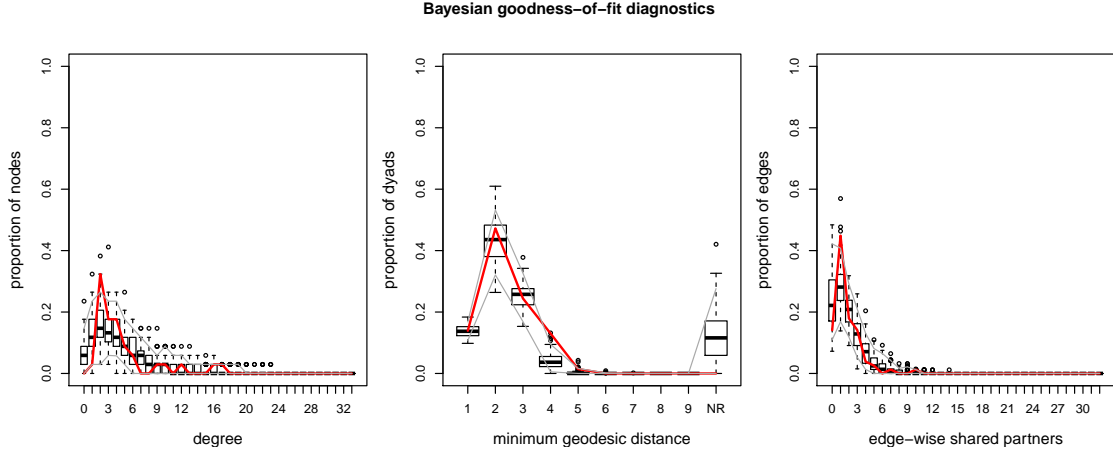


Figure 3.12 Bayesian goodness-of-fit diagnostics for the mixed model with GWESP (with fixed decay of 0.8) and nodal random effects for the karate club data. Bold red line corresponds to original dataset.²

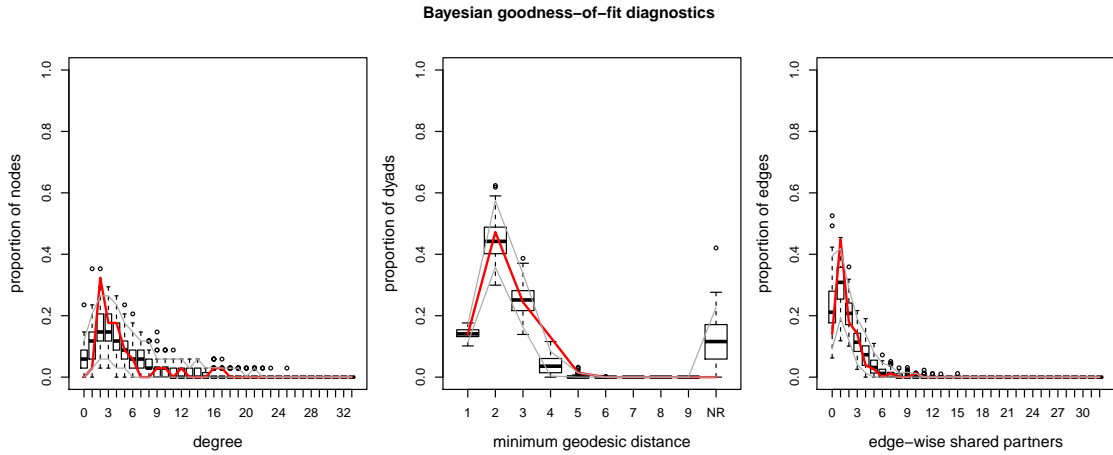


Figure 3.13 Bayesian goodness-of-fit diagnostics for model with random effects only for the karate club data. Bold red line corresponds to original dataset.²

Kapferer Tailor Shop

As a second data example we use the Kapferer network (Kapferer, 1972) which contains interactions among 39 workers in a tailor shop in Zambia. Figure 3.14 shows a plot of the network. The situation here is comparable to the karate data with only some high degree nodes.

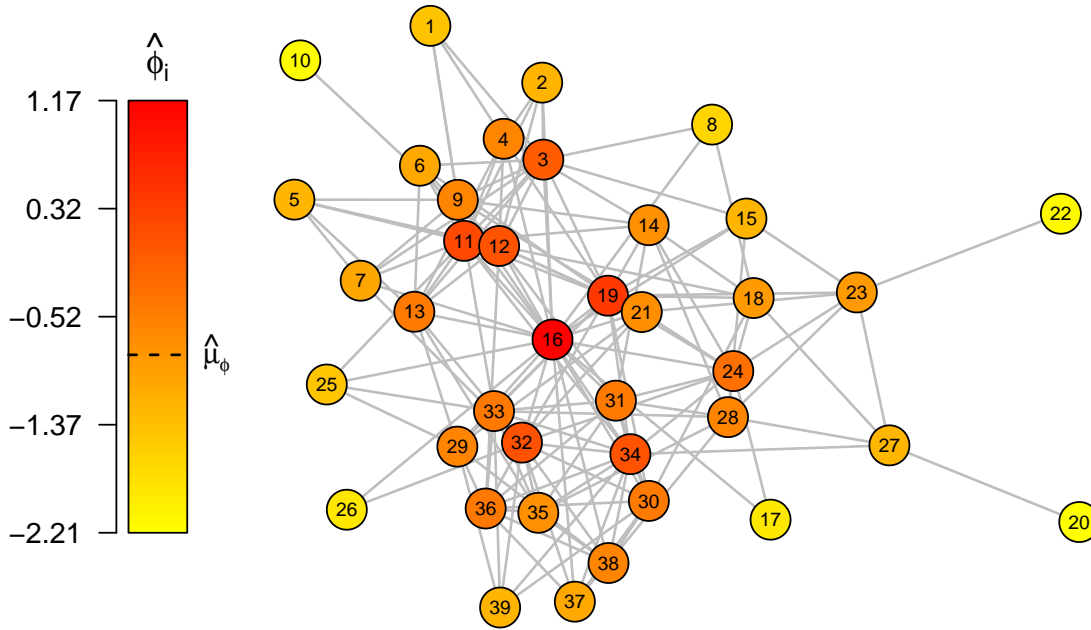


Figure 3.14 *Kapferer Tailor Shop Network*. Vertices are coloured by their estimated nodal effect $\hat{\phi}_i$ (posterior mean), $i = 1, \dots, n$. Vertices with a high nodal effect are darker in orange/red.

Following Robins and Lusher (2013) we start with a fixed model containing edges, geometrically weighted degree with a fixed decay parameter of 0.7 and the 2-star statistic. The geometrically weighted degree statistic has the form

$$e^{\theta_{\text{dec}}} \sum_{i=1}^{n-1} \left\{ 1 - \left(1 - e^{-\theta_{\text{dec}}} \right)^i \right\} D_i(\mathbf{y}),$$

where $D_i(\mathbf{y})$ denotes the number of nodes with degree i , see Hunter (2007) for details. Again, we set the decay parameter to a fixed value $\theta_{\text{dec}} = 0.7$ resulting in a regular, i.e. a non-curved ERGM (Hunter and Handcock, 2006). We fit a Bayesian Exponential Random Graph Model with 6 chains and adaptive direction sampling (Caimo and Friel, 2011) in

Table 3.3 *Model fitting results for the Kapferer data. The fixed model contains edges, geometrically weighted degree (with a fixed decay of 0.7), and the 2-star effect. The random model contains only the nodal random effects.*

Model type	Parameter	Post. mean	Post. Sd.	Acceptance rate	Note
fixed	θ_{edges}	-2.59	0.41		
	$\theta_{\text{gwdegreeFixed0.7}}$	-0.30	1.18	0.03	
	$\theta_{\text{2-star}}$	0.08	0.02		
random	μ_{ϕ}	-0.82	0.16	0.22	
	σ_{ϕ}^2	0.82	0.31	0.43	*

* For σ_{ϕ}^2 the posterior mean is calculated based on the logarithmized values and then transformed back to the scale of σ_{ϕ}^2 (this leads to the geometric mean) due to the non-symmetric posterior density in this case.

order to improve mixing and obtain a more stable result than for a single chain model. The results for the overall chain are shown in Figure 3.15. Table 3.3 summarizes the results. The overall tendency of a small positive 2-star effect combined with a negative effect of the geometrically weighted degree corresponds to the suggestions of Robins and Lusher (2013) for modelling a heterogeneous degree distribution with some high degree nodes.

As competing model we fit a Bayesian ERGM with nodal random effects only, which is again basically the Bayesian analogue of a p_2 model. The results are shown in Figure 3.16 and Table 3.3.

Figure 3.17 and Figure 3.18 show Bayesian goodness-of-fit plots for both models. Similar to the goodness-of-fit plots in the previous data example, for each model we used 100 draws from the corresponding posterior and simulated a network for each of the posterior parameter combinations. Again, the plots show boxplots of the distributions of degree, minimum geodesic distance, and edge-wise shared partners for the resulting simulated networks. The bold red line indicates the values of the original Kapferer network. Degenerate behaviour did not occur here, i.e. there were no networks with high proportions of nodes with degree 38 or edges with 37 shared partners. Note that therefore the plots do not show the full distributions of the three measures to ease visual inspection. For the model with nodal random effects only the resulting simulated networks seem to be less extreme in the sense that the observed values for all three statistics are closer to the simulated ones and the boxplots are not as wide as for the fixed model.

The resulting log Bayes factor for the two non-nested models is 217, which is again quite big and supports the model with random effects only.

Including both effects simultaneously into the model, that means fitting a BERGM with

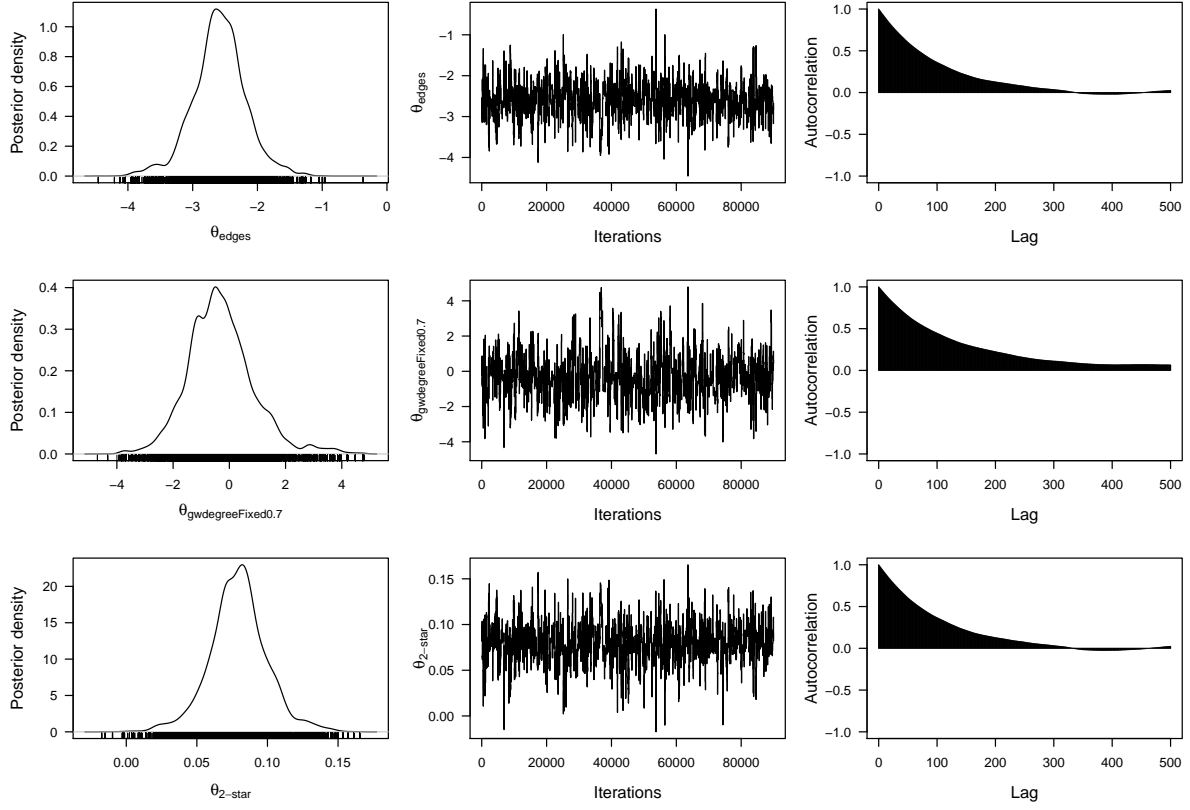


Figure 3.15 *Posterior densities, trace plots, and autocorrelation for the fixed model with edges, geometrically weighted degree (with fixed decay of 0.7), and 2-star effect for the Kapferer data.*

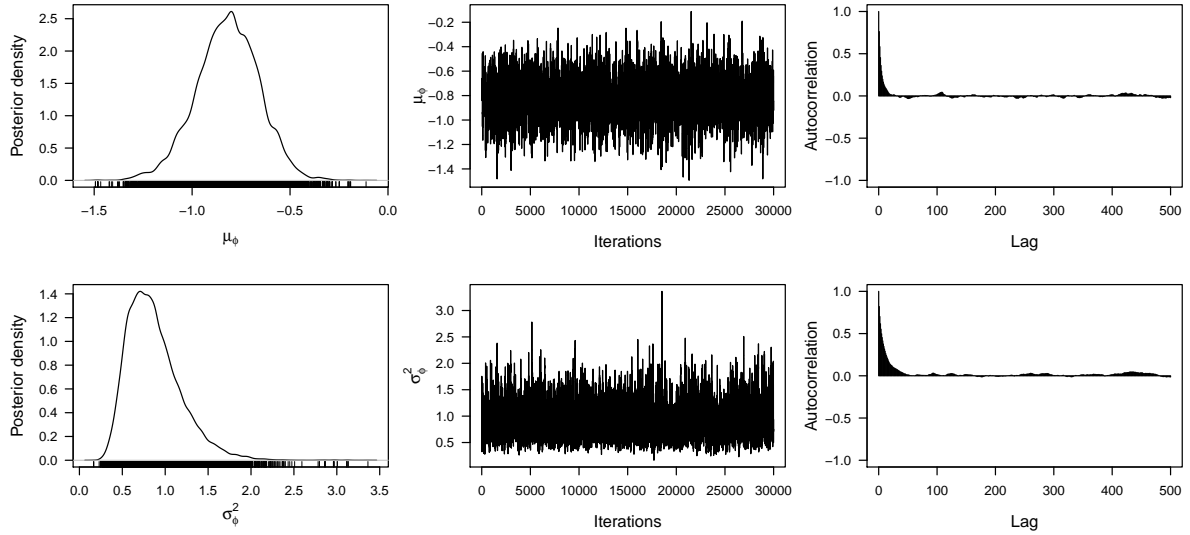


Figure 3.16 *Posterior densities, trace plots, and autocorrelation for the model with nodal random effects for the Kapferer data.*

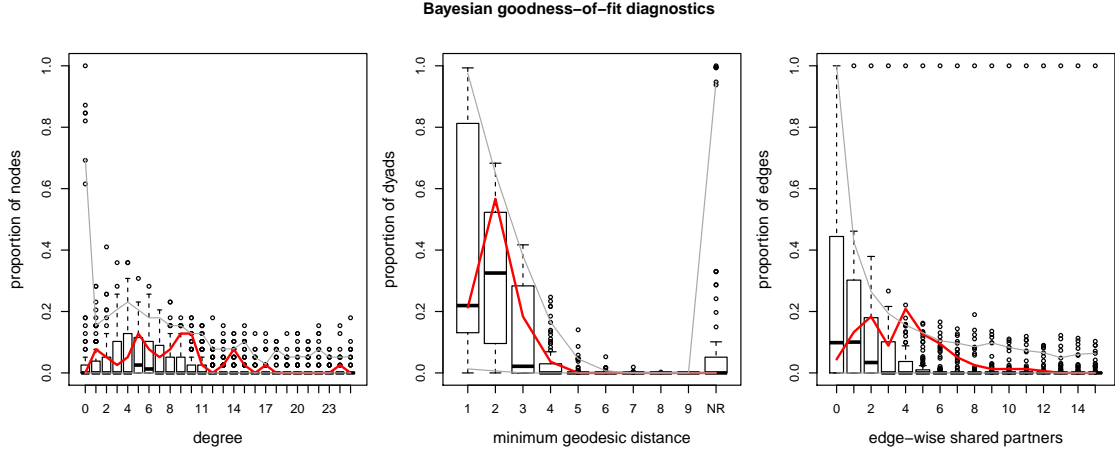


Figure 3.17 *Bayesian goodness-of-fit diagnostics for the fixed model with edges, geometrically weighted degree (with fixed decay of 0.7), and 2-star effect for the Kapferer data. Bold red line corresponds to original dataset.*

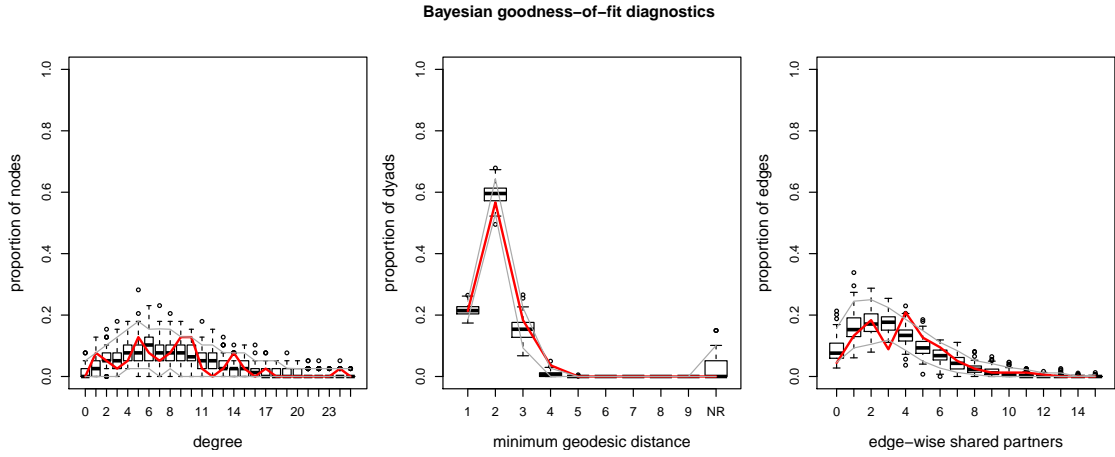


Figure 3.18 *Bayesian goodness-of-fit diagnostics for the model with nodal random effects for the Kapferer data. Bold red line corresponds to original dataset.*

geometrically weighted degree and nodal random effects, would technically be possible, but is not advisable from our experience. The resulting MCMC runs are quite unstable, meaning we get very low acceptance probabilities, and bad mixing behaviour of the chain. Both, geometrically weighted degree and the nodal random effects, are based on the degree of the nodes and we suppose that this leads to some identifiability issues which cause this behaviour.

European Parliament Members

The third data example consists of a network of members of the European parliament (MEP) in of the 6th legislative period. The complete network contains more than 900 vertices. We analyse a subset of the 32 members from the Netherlands. The induced subgraph is shown in Figure 3.19. A link between to MEPs exists if they have at least one committee membership in common. The data were provided by Paul W. Thurner (see Thurner et al., 2013). This data example illustrates our model selection procedure and clearly indicates that not always the more complicated model with more parameters is preferred.

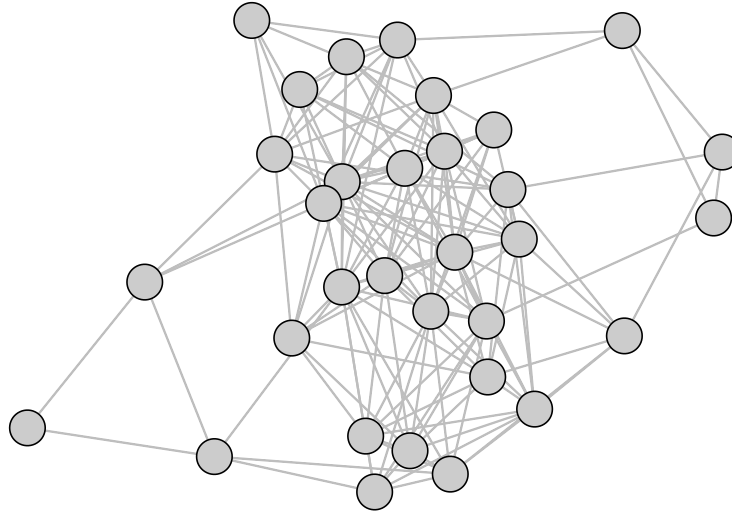


Figure 3.19 *Network of Dutch members of the European parliament during the 6th legislative period. Two members are linked if they have at least one committee membership in common.*

We fitted the same two nested models to the data as in the first part of the karate club example: a standard ERGM with edges and triangles as sufficient statistics, and a model with nodal random effects and the triangle statistic. The number of iterations was also equivalent.

Figures 3.20 and 3.21 show the results, which are also summarised in Table 3.4. For the mixed model we get a very low acceptance rate for the triangle effect and very high autocorrelations for the triangle effect and the mean parameter μ_ϕ . The later could possibly be solved by thinning out the chain.

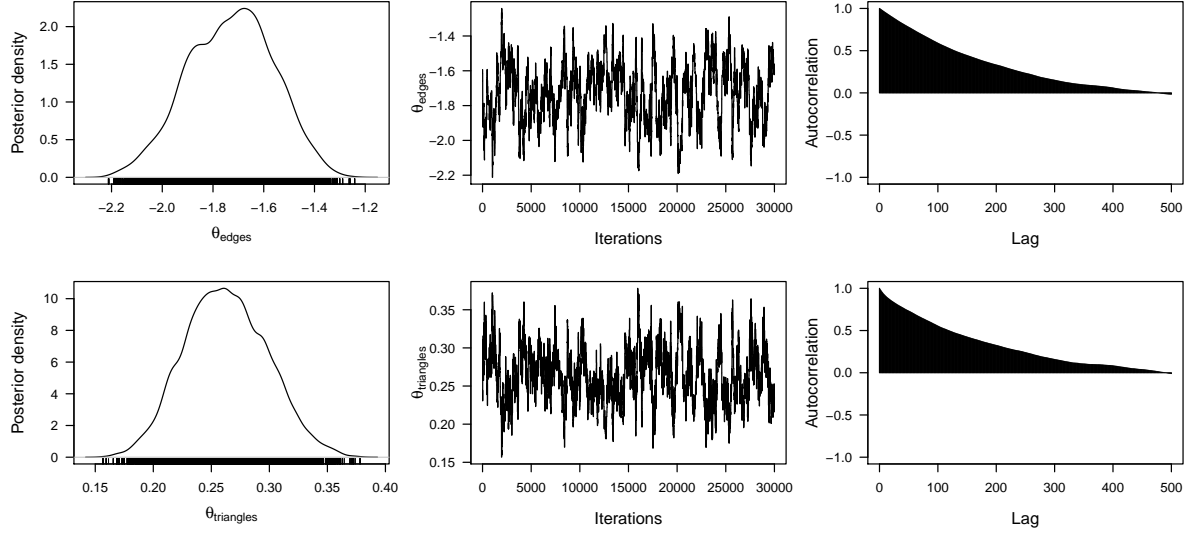


Figure 3.20 Posterior densities, trace plots, and autocorrelation for the fixed model with edges and triangular effect for the European parliament data.

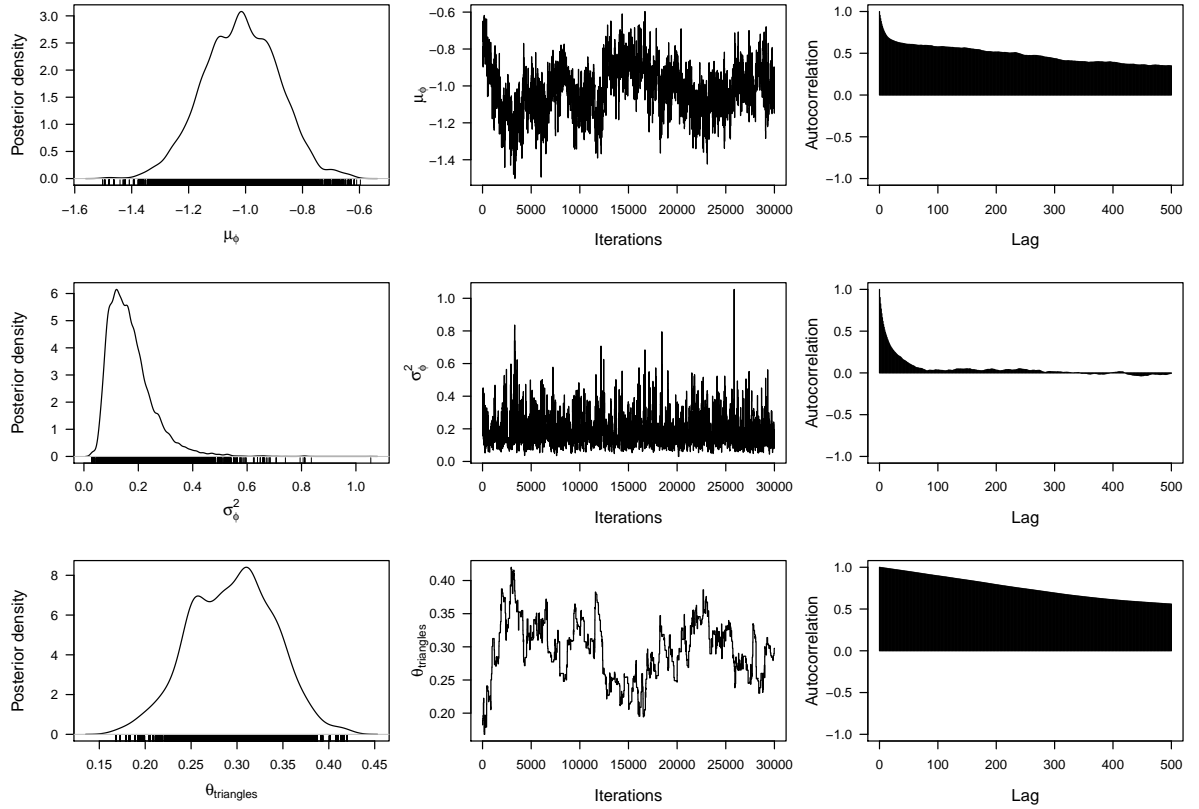


Figure 3.21 Posterior densities, trace plots, and autocorrelation for the mixed model with nodal random and triangular effects for the European parliament data.

Table 3.4 *Model fitting results for the European parliament data. The fixed model contains edges and triangles, and the mixed model triangles and nodal random effects.*

Model type	Parameter	Post. mean	Post. Sd.	Acceptance rate	Note
fixed	θ_{edges}	-1.73	0.17	0.13	
	$\theta_{\text{triangles}}$	0.26	0.04		
mixed	μ_ϕ	-1.02	0.13	0.11	*
	σ_ϕ^2	0.15	0.08	0.15	
	$\theta_{\text{triangles}}$	0.29	0.05	0.02	

* For σ_ϕ^2 the posterior mean is calculated based on the logarithmized values and then transformed back to the scale of σ_ϕ^2 (this leads to the geometric mean) due to the non-symmetric posterior density in this case.

Nevertheless, the focus in this example is on model selection. The computed log Bayes factor is -13.9 and clearly indicates that the model without nodal random effects is preferable in this situation. Apparently, here we have a network dataset where there is no benefit in including nodal random effects into the model. This corresponds to the rather small estimate for the variance of the nodal random effects σ_ϕ^2 . The resulting Bayes factor shows that it is not the case that the model with more parameters is always selected. This can also be seen from the simulation results in the following subsection.

3.4.2 Simulation

For the simulation study we used the following components based on two very simple, but different model generating processes, a nodal random effects only situation, i.e. the p_2 model, and structural effects only situation, i.e. the classical ERGM. For each setting we generated networks with 40 vertices, using again the simulation routines from the `ergm` package (Hunter et al., 2008b). The first model (A) was the one with nodal random effects only, i.e.

$$\begin{aligned} \text{logit}\left[\mathbb{P}\left(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\phi}\right)\right] &= \phi_i + \phi_j, \\ \text{with } \phi_i &\sim N\left(\mu_\phi, \sigma_\phi^2\right), \quad \text{for } i = 1, \dots, n. \end{aligned} \tag{3.17}$$

The parameter μ_ϕ was constantly set to $\mu_\phi = -1$, so that the resulting network graphs tend to be rather sparse. For σ_ϕ^2 we used values between 0 and 1. Model (B) was the standard ERGM with edges and 2-star statistics, and no nodal random effects, i.e. $\boldsymbol{\theta} = \left(\theta_{\text{edges}}, \theta_{2\text{-star}}\right)^t$

and

$$\text{logit}\left[\mathbb{P}\left(Y_{ij} = 1 | Y_{kl}, (k, l) \neq (i, j); \boldsymbol{\theta}\right)\right] = \theta_{\text{edges}} + \theta_{2\text{-star}} \cdot \left[\sum_{k \neq j} y_{ik} + \sum_{l \neq i} y_{jl} \right]. \quad (3.18)$$

The parameter θ_{edges} was constantly set to $\theta_{\text{edges}} = -2$. This is equivalent to model (A) in the sense that $2 \cdot \mu_{\phi} = \theta_{\text{edges}}$, because θ_{edges} is a parameter on a per link basis, μ_{ϕ} is on a per node basis and one needs two nodes to form a link. For $\theta_{2\text{-star}}$ we used values between 0 and 0.05. This value needs to be small, i.e. close to zero, because otherwise we only generate full, or empty graphs if the value is negative, see also Schweinberger (2011).

For each of the resulting parameter combinations in model (A) and model (B) we generated 50 networks.

For the chosen settings the resulting 40 node networks seem to be reasonable. We get an average network density between 0.11 and 0.30 for the different settings.

Note that setting $\sigma_{\phi}^2 = 0$ in model (A) and $\theta_{2\text{-star}} = 0$ in model (B) leads to a simple

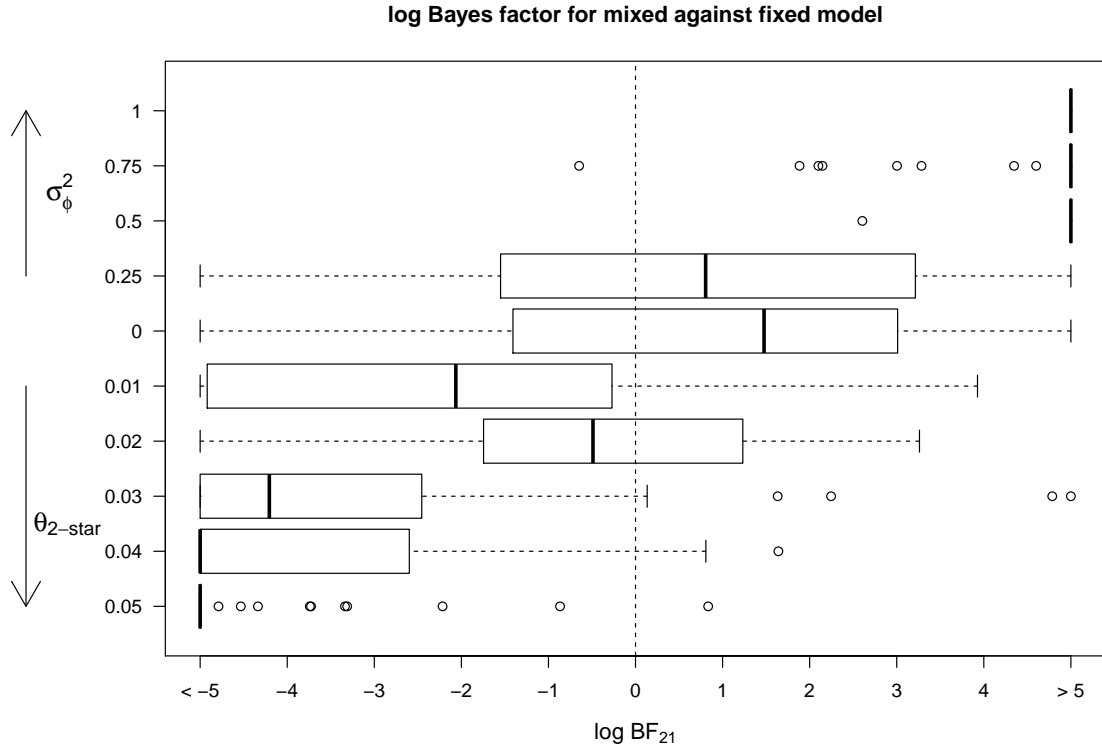


Figure 3.22 Resulting log Bayes factors for the mixed model against the fixed model (nested models) for different simulation settings. The annotation on the y-axis shows which was the underlying true model, a model with nodal random effects only in the direction of σ_{ϕ}^2 , and a model with edges and 2-stars in the direction of $\theta_{2\text{-star}}$.

Table 3.5 Resulting Bays factors (mixed model against fixed model, nested models) for the simulation from setting (A) a nodal random effects only situation, and setting (B) a classical ERGM with edges and 2-star statistics. Each setting was run 50 times, except for the Bernoulli setting, which had $2 \cdot 50$ runs.

Setting		average nw density	log Bayes factor for mixed against fixed model					
			min	max	% < -5	% < 0	% > 0	% > 5
(A) random effects	$\sigma_\phi^2 = 1$	0.23	13.03	137.64	0	0	100	100
	$\sigma_\phi^2 = 0.75$	0.11	-0.65	498.53	0	2	98	84
	$\sigma_\phi^2 = 0.5$	0.16	2.60	350.05	0	0	100	98
	$\sigma_\phi^2 = 0.25$	0.15	-7.73	292.34	6	34	66	20
Bernoulli network								
	$\sigma_\phi^2 = \theta_{2\text{-star}} = 0$	0.13	-7.80	10.27	4	37	63	4
(B) fixed effects	$\theta_{2\text{-star}} = 0.01$	0.13	-14.56	3.93	24	76	24	0
	$\theta_{2\text{-star}} = 0.02$	0.14	-144.23	3.26	10	54	46	0
	$\theta_{2\text{-star}} = 0.03$	0.16	-25.24	51.34	44	88	12	2
	$\theta_{2\text{-star}} = 0.04$	0.20	-240.76	1.64	64	94	6	0
	$\theta_{2\text{-star}} = 0.05$	0.30	-218.07	0.83	80	98	2	0

Note: For setting (A) we set $\mu_\phi = -1$, and for setting (B) $\theta_{\text{edges}} = -2$, so that $\mu_\phi = 2 \cdot \theta_{\text{edges}}$.

Bernoulli network, which can be seen as a null model.

As a first step, similarly to the karate data example, we fitted two nested models to each of the simulated networks: a standard ERGM with edges and 2-stars as sufficient statistics, and a model with nodal random effects and the 2-star statistic. Again this step was followed by computing a Bayes factor to compare the model with nodal random effects to the one with structural effects only.

Figure 3.22 shows boxplots of the resulting log Bayes factors for the different settings. For the plot the log Bayes factors were cut at values of -5 and 5 because some were really small or really large. These cutting values were chosen following Kass and Raftery (1995). More detailed information, especially on the range of the simulation results is given in Table 3.5. For the null model of a pure Bernoulli network the log Bayes factor can point in either one of the directions, the same is more or less true for only small deviations from this null model. The general impression is, that the more extreme the underlying setting becomes the sooner the log Bayes factor points into the correct direction.

Most importantly the results of the simulation show that our model selection works with respect to the size of the competing models. It is not the case that the model with more parameters, which is the model with nodal random effects, is always preferred.

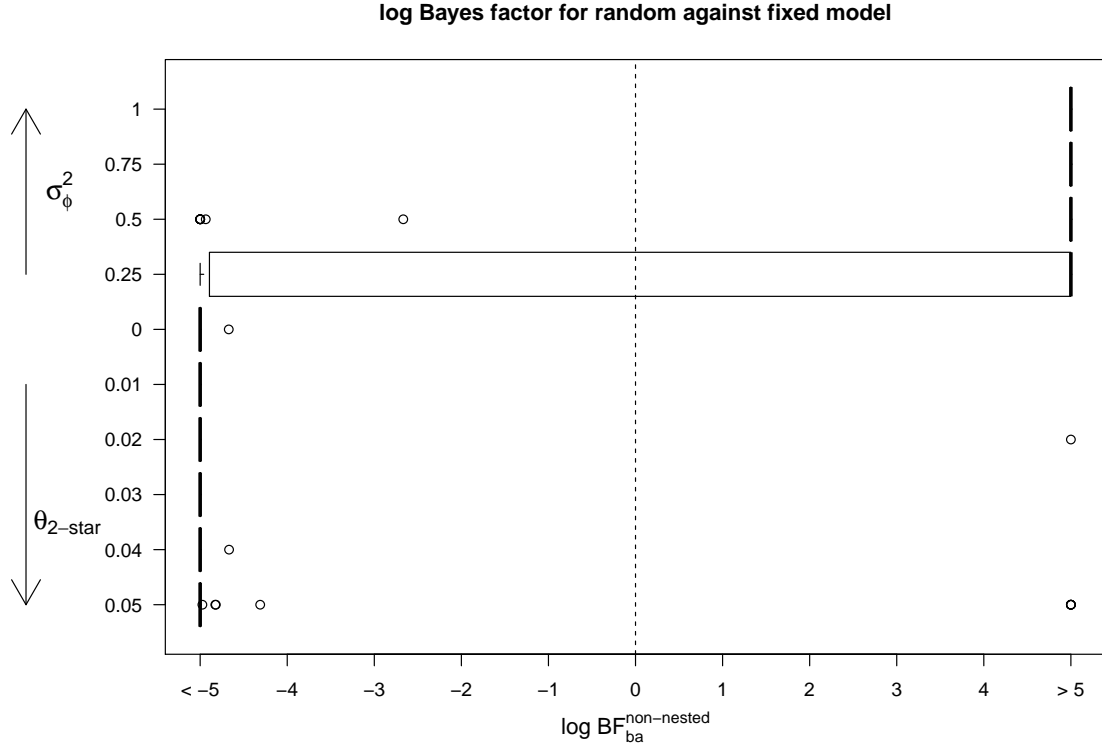


Figure 3.23 Resulting log Bayes factors for model a with random effects only against model b with fixed effects only (non-nested models) for different simulation settings. The annotation on the y -axis shows which was the underlying true model, a model with nodal random effects only in the direction of σ_ϕ^2 , and a model with edges and 2-stars in the direction of $\theta_{2\text{-star}}$.

Additionally based on the same simulated networks we compared two non-nested models: the standard ERGM with edges and 2-stars from before, and a model with nodal random effects only. In this case we computed a log Bayes factor for non-nested models as described in Section 3.3.3. Figure 3.23 shows the results. Here the distinction seems to be clearer, i.e. the log Bayes factor in most cases clearly indicates the correct direction. In case of the null model the standard ERGM is preferred, i.e. the smaller model which has fewer parameters.

3.5 Discussion and Summary

Heterogeneity of actors in the network is usually modelled by including (known) nodal or bi-nodal covariates, see, e.g., Robins et al. (2001). Latent, random heterogeneity is considered only exceptionally, for example in Krivitsky et al. (2009), who implicitly assume that the local structure of the network is homogeneous. In particular, this implies that well studied phenomena, such as a small-world networks, Milgram (1967), Watts and Strogatz (1998), where shortest path lengths between two nodes in the network tend to be very small and scale-free networks, where few nodes have unusually high degree, are not appropriately modelled using the standard statistical modelling approaches. This is particularly true for Exponential Random Graph Models.

Here our extension of the Exponential Random Graph Model (ERGM) avoids the assumption of nodal homogeneity. By adding nodal random effects to the model we get a flexible tool to model heterogeneity in the network which is not captured in available (nodal) covariates otherwise. Using the Bayesian framework for ERGMs proposed by Caimo and Friel (2011) allows us to add this random effects extension to the model in an elegant and rather straightforward manner. Estimating Bayes factors enables us to handle the problem of model selection associated with this modelling task. The resulting estimates for the three data examples seem to be reasonable.

Furthermore, the small simulation study in the previous section suggests that in general the Bayes factor approach seems to work and even though a mixed model with nodal random effects has more parameters than its fixed equivalent it is not systematically preferred in the model selection.

The model can at least conceptually be extended to allow for node specific network effects. In this case one replaces the right hand side in model (3.5) through $\boldsymbol{\theta}^t s_{ij}(\mathbf{y}) + \tilde{s}_{ij}(\mathbf{y})(\phi_i + \phi_j)$, where $\tilde{s}_{ij}(\mathbf{y})$ is a subvector of $s_{ij}(\mathbf{y})$. In this case one may, for instance, model that the 2-star effect is heterogeneous amongst the nodes in the network. However, even though conceptually this is straightforward, the computation in the Bayesian estimation is even more challenging.

We should note that the approach which we have introduced is computationally intensive. A promising avenue of research to address this issue is to explore approximations of the likelihood function using composite likelihoods, of which the Pseudo-Likelihood approximation of Frank and Strauss (1986) is an antecedent. We refer the reader to Varin et al. (2011) for a recent review of composite likelihoods. We are currently engaged in work in this direction.

Secondly, as parallel computing is becoming more accessible this will help to shorten the time for computation. This also applies to parallel computing in the Gibbs step of the algorithm.

Acknowledgements

We gratefully acknowledge the data provision by Paul W. Thurner.

The Insight Centre for Data Analytics is supported by Science Foundation Ireland under Grant Number SFI/12/RC/2289. Nial Friel's research was also supported by an Science Foundation Ireland grant: 12/IP/1424.

Alberto Caimo's research was supported by the Swiss National Science Foundation (SNSF), under grant: CR12I1-156229.

4 Stable Exponential Random Graph Models with Non-parametric Components for Large Dense Networks

Abstract

Exponential Random Graph Models (ERGM) behave peculiar in large networks with thousand(s) of actors (nodes). Standard models containing 2-star or triangle counts as statistics are often unstable leading to completely full or empty networks. Moreover, numerical methods break down which makes it complicated to apply ERGMs to large networks. In this chapter we propose two strategies to circumvent these obstacles. First, we fit a model to a subsampled network and secondly, we show how linear statistics (like 2-stars etc.) can be replaced by smooth functional components. These two steps in combination allow to fit stable models to large network data, which is illustrated by a data example including a residual analysis.

Contributed Manuscript

This chapter is in most parts equivalent to a submitted manuscript, which is currently under review and available on arXiv as online pre-print,

Thiemichen, S. and Kauermann, G. (2016). Stable exponential random graph models with non-parametric components for large dense networks.
arXiv preprint arXiv:1604.04732.

except for a few corrections, mainly concerning orthography, and small adjustments as the paper serves as a chapter in this thesis and no longer as stand-alone article.

This is joint work with Göran Kauermann (Institut für Statistik, Ludwigs-Maximilians-Universität München, Germany). The basic ideas for the subsampling scheme based on Latin Squares came from Göran Kauermann. Both authors developed the idea of including smooth functional components into Exponential Random Graph Models.

Stephanie Thiemichen wrote the algorithmic implementation for the subsampling and the non-parametric extension, and conducted the data analysis including the residual analysis. Both authors contributed to the concrete elaboration of the model extension, wrote the manuscript, and were involved in proof-reading.

Software

All computations and most plots in this chapter have been produced using R version 3.2.3 with packages `fda` 2.4.4, `Matrix` 1.2-3, `ergm` 3.5.1, `network` 1.13.0, `statnet.common` 3.3.0, and `quadprog` 1.5-5. For parallelisation R's base package `parallel` was used.

Our algorithms for subsampling and model fitting are available in the package `ergam` on github (<https://github.com/sthiemichen/ergam>).

The visualisation of the Facebook network data example in Figure 4.4 has been generated using `visone` (version 2.16).

4.1 Introduction

The analysis of network data is an emerging field in statistics. It is challenging both model-wise and computationally. Recently, Goldenberg et al. (2010), Hunter et al. (2012), and Fienberg (2012) published comprehensive survey articles discussing new statistical approaches and developments in network data analysis. We also refer to the monograph of Kolaczyk (2009) for a general introduction to the field, or the recent book of Lusher et al. (2013), which focuses on a specific and widely used class of network models, so-called Exponential Random Graph Models (ERGM).

In its most simple form a network consists of a set of n nodes (actors) which are potentially linked with each other through edges. These edges between the actors are thereby the focus of interest. Notationally a network can be expressed as a $n \times n$ (random) adjacency matrix \mathbf{Y} with entries $Y_{ij} = 1$ if node i and j are connected, and $Y_{ij} = 0$ otherwise. In undirected networks one has $Y_{ij} = Y_{ji}$ while for directed links we have $Y_{ij} = 1$ if a directed edge goes from node i to node j . For the sake of readability and notional simplicity we will concentrate here on undirected networks. The term \mathbf{y} denotes a concrete realisation of \mathbf{Y} .

A common and powerful model for network data \mathbf{Y} was proposed by Frank and Strauss (1986) as Exponential Random Graph Model (ERGM) taking the form

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \boldsymbol{\theta}) = \frac{\exp \left\{ \sum_{l=0}^p s_l(\mathbf{y}) \theta_l \right\}}{\kappa(\boldsymbol{\theta})}, \quad (4.1)$$

with $\boldsymbol{\theta} = (\theta_0, \dots, \theta_p)^t$ as parameter vector and $s(\mathbf{y}) = (s_0(\mathbf{y}), \dots, s_p(\mathbf{y}))^t$ as vector of statistics of the network. In equation (4.1) the term $\kappa(\boldsymbol{\theta})$ denotes the normalizing constant, that is

$$\kappa(\boldsymbol{\theta}) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp \left\{ \boldsymbol{\theta}^t s(\mathbf{y}) \right\},$$

where \mathcal{Y} is the set of all networks and accordingly the sum is over $2^{\binom{n}{2}}$ terms. It is therefore numerically intractable, except for very small graphs. We denote with $s_0(\mathbf{y}) = \sum_{i=1}^n \sum_{j>i}^n y_{ij}$ the baseline statistic giving the number of edges in the (undirected) network, so that θ_0 serves as intercept. The interpretation of the remaining parameters θ_l , $l = 1, \dots, p$, results through the corresponding conditional model for each single edge Y_{ij} given the remaining network $\mathbf{Y} \setminus Y_{ij}$, since

$$\text{logit} \left[\mathbb{P}(Y_{ij} = 1 | \mathbf{Y} \setminus Y_{ij}; \boldsymbol{\theta}) \right] = \theta_0 + \sum_{l=1}^p \Delta_{ij} s_l(\mathbf{y}) \theta_l, \quad (4.2)$$

where $\Delta_{ij}s_l(\mathbf{y}) = s_l(\mathbf{y} \setminus y_{ij}, y_{ij} = 1) - s_l(\mathbf{y} \setminus y_{ij}, y_{ij} = 0)$ is the so-called change statistics which is obtained by flipping the edge between nodes i and j from non-existent to existent.

Exponential Random Graph Models are numerically unstable, in particular if the number of actors n gets large. Hence, for large networks one is faced with two relevant problems. First, the model itself is notoriously unstable leading to either full or empty networks. This issue is usually called degeneracy problem, see, for example, (Schweinberger, 2011), and Chatterjee and Diaconis (2013). Secondly, the estimation is per se numerically demanding or even unfeasible since numerical simulation routines are too time consuming. We aim to tackle both problems in this paper. First, we propose the use of stable statistics which are derived as smooth, non-parametric curves. Secondly, instead of fitting the model to the entire network we propose to draw samples from the network such that estimation in each sample is numerically (very) easy. These two proposals allow to easily analyse network data in large and sufficiently dense networks.

Schweinberger (2011) denotes network statistics (and the corresponding ERGM) as unstable if the statistics is not at least of order $O_p(n)$. In fact he shows that any k -star or triangle statistics is unstable leading to an odd behaviour of model (4.1). Effectively, unstable networks are either complete (i.e. have all possible edges) or empty (i.e. all nodes are unconnected) unless for a diminishing subspace of the parameter space for n increasing. If n gets large it is therefore advisable to replace the statistics in model (4.1) by stable statistics of order $O_p(n)$. A first proposal in this direction are alternating star and alternating triangle statistics as proposed in Snijders et al. (2006), or geometrically weighted statistics as proposed in the context of Curved Exponential Random Graph Models, see Hunter and Handcock (2006). Hunter (2007) shows that from a modelling point of view the alternating statistics are equivalent to geometrically weighted degree or geometrically weighted edge-wise shared partners, respectively. Both approaches stabilize the models but for the price of less intuitive interpretations of the parameter estimates. We propose an alternative by making use of non-parametric models based and the technique of smoothing (see, e.g., Ruppert et al., 2003). The non-parametric model thereby maintains the interpretability of the ERGM based on the conditional model (4.2). To motivate our idea we start with the conditional model (4.2) and replace the linear terms through non-linear smooth components. This leads to the conditional non-parametric model

$$\text{logit} \left[\mathbb{P}(Y_{ij} = 1 | \mathbf{Y} \setminus Y_{ij}) \right] = \theta_0 + \sum_{l=1}^p m_l(\Delta_{ij}s_l(\mathbf{y})), \quad (4.3)$$

where $m_l(\cdot)$ are smooth functions which need to be estimated from the data. Models of type (4.3) have been proposed in a simple regression framework as generalized additive models, see, e.g., Hastie and Tibshirani (1990), or Wood (2006), but apparently the structure here

is more complex as we are tackling network data. We additionally need to postulate that functions $m_l(\cdot)$ are monotone and bounded which in turn leads to stable network statistics in the definition of Schweinberger (2011). We make use of penalized spline smoothing which also allows to accommodate constraints on the functional shape leading to stable network models. In fact, assuming $m_l(\cdot)$ to be monotone and bounded, we may derive a non-parametric Exponential Random Graph Model from (4.3) which takes the form

$$\mathbb{P}(\mathbf{Y} = \mathbf{y} | \theta_0, m_l(\cdot), l = 1, \dots, p) = \frac{\exp \left\{ s_0(\mathbf{y})\theta_0 + \sum_{l=1}^p \sum_i \sum_{j>i} y_{ij} m_l(\Delta_{ij} s_l(\mathbf{y})) \right\}}{\kappa(\theta_0, m_l(\cdot), l = 1, \dots, p)} \quad (4.4)$$

Apparently, model (4.4) appears rather complex due to its semi-parametric structure and estimation looks like a challenging task. We will argue, however, that smoothing techniques can easily be applied and estimation becomes feasible by making use of sampling strategies in networks leading to numerically simple likelihoods and in fact consistent (though not efficient) estimates.

Estimation in Exponential Random Graph Models is cumbersome and numerically demanding as it requires simulation based routines. Snijders (2002) suggests the calculation of $\partial \kappa(\boldsymbol{\theta}) / \partial \boldsymbol{\theta}$ in the score equation resulting from (4.1) using stochastic approximation. Hunter and Handcock (2006) propose to use MCMC methods in order to obtain the maximum likelihood estimate. The approach is extended and improved in Hummel et al. (2012). In a recent paper Caimo and Friel (2011) develop a fully Bayesian estimation routine by incorporating the so-called exchange algorithm from Murray et al. (2006) which circumvents the calculation or approximation of the normalisation constant for the price of extended MCMC sampling. A general survey of available routines for fitting Exponential Random Graph Models is given in Hunter et al. (2012). In fact, if the network is large, MCMC based routines readily become numerically infeasible. As aforementioned, we will therefore make use of subsampling the network data and fit the model to subsamples that allow for simple likelihoods. We follow ideas of Koskinen and Daraganova (2013). In fact, for models with k -stars or triangles only, the edges follow a Markovian independence structure by conditioning on parts of the network (see Frank and Strauss, 1986, or Whittaker, 2009). This is exemplified in a simple network with four nodes in Figure 4.1. Conditioning on edges Y_{12}, Y_{14}, Y_{23} , and Y_{34} we find that Y_{13} and Y_{24} are conditionally independent, which can be denoted as $Y_{13} \perp\!\!\!\perp Y_{24} | \mathbf{Y} \setminus \{Y_{13}, Y_{24}\}$. The idea is now to make use of this independence property to fit model (4.4) to a subsample of the network while conditioning on the rest of the network. Hence, exemplary we sample edges Y_{13} and Y_{24} , and condition on $\mathbf{Y} \setminus \{Y_{13}, Y_{24}\}$. Due to the (conditional) independence structure we can easily fit the conditional model (4.2) with standard software for generalized linear and non-parametric additive models. This will be demonstrated below. Apparently

such a strategy is not efficient if the network is small, but if the network is (very) large and (sufficiently) dense, sampling appears as a plausible approach which also maintains numerical feasibility.

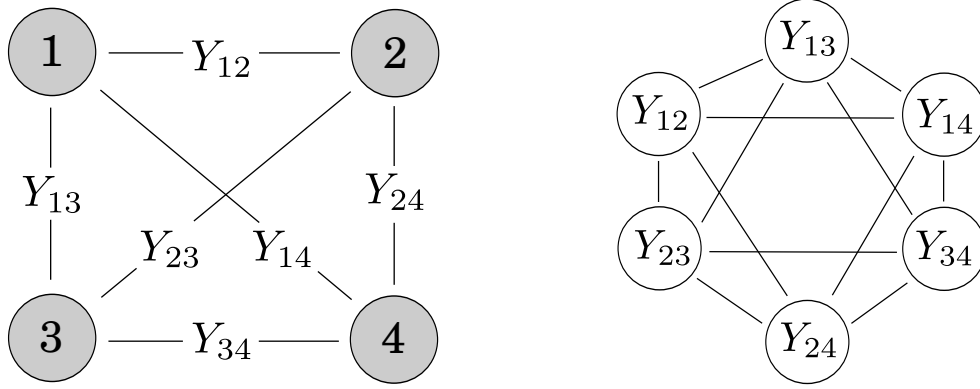


Figure 4.1 Visualisation of the induced Markov independence graph (right) for an Exponential Random Graph Model for a simple 4-node network (left).

The chapter is organized as follows. In Section 4.2 we suggest to estimate large Exponential Random Graph Models through subsampling of the network. In Section 4.3 we extend the idea towards non-parametric models. Section 4.4 gives a data example demonstrating the usability of the approach. Finally, a discussion completes the chapter in Section 4.5.

All routines for fitting and analysing the models are written in R (R Core Team, 2016) and are available as R package `ergam` on github (<https://github.com/sthiemichen/ergam>).

4.2 Estimation through Subsampling

The general idea proposed in this section is that instead of fitting an ERGM to the entire data, we fit a conditional model to appropriate subsamples of the data. Due to conditional independence this allows for fast and easy computing. We start the presentation with the classical (unstable) ERGM and assume model (4.1) has statistics like k -star and triangle effects only. That is statistics $s_l(\cdot)$ in (4.1) for instance has no “4-cycles” of the form $\sum_{i < j < k < l} Y_{ij} Y_{jk} Y_{kl} Y_{li}$ (or higher order cycles). Let us get more specific. For simplicity of presentation let n , the number of nodes in the network, be even. With $\mathcal{D}(n|2)$ we denote a decomposition of the set $\{1, \dots, n\}$ into subsets of size 2, e.g., $\mathcal{D}(n|2) = \{(1, 2), (3, 4), \dots, (n-1, n)\}$. For $A = (i, j) \in \mathcal{D}(n|2)$ we denote $Y_A = Y_{ij}$ and $\mathbf{Y} \setminus Y_{\mathcal{D}(n|2)} = \{Y_{ij}, (i, j) \notin \mathcal{D}(n|2)\}$. Apparently $\mathcal{D}(n|2)$ has $n/2$ elements. We assume

now that the statistics $s_l(\mathbf{y})$ in (4.1) can be decomposed to

$$s_l(\mathbf{y}) = \sum_{A \in \mathcal{D}(n|2)} s_{lA}(y_A, \mathbf{y} \setminus y_{\mathcal{D}(n|2)}). \quad (4.5)$$

This holds for all k -stars and triangle statistics. It is not difficult to show that with condition (4.5) density (4.1) can then be factorized to

$$\mathbb{P}(\mathbf{Y} = \mathbf{y}) = \prod_{A \in \mathcal{D}(n|2)} h_A(y_A, \mathbf{y} \setminus y_{\mathcal{D}(n|2)}), \quad (4.6)$$

where $h_A(\cdot)$ is some function depending on $\{y_A, \mathbf{y} \setminus y_{\mathcal{D}(n|2)}\}$. The factorization (4.6) implies that the edges with indices in $\mathcal{D}(n|2)$ are mutually independent conditional on the rest of the network (see Whittaker, 2009).

The conditional independence will be used to fit model (4.1) not for the entire network but for an appropriately chosen subnetwork. We therefore draw a sample of the network \mathbf{Y} by taking \mathbf{y}_A with $A \in \mathcal{D}(n|2)$ as sampled binary observations accompanied by $\Delta_A s(\mathbf{y}) = (\Delta_A s_1(\mathbf{y}), \dots, \Delta_A s_p(\mathbf{y}))^t$ as corresponding change statistics with obvious definition of $\Delta_A s(\cdot)$. The term $\Delta_A s(\mathbf{y})$ plays the role of covariates and the conditional model (4.2) takes the form

$$\text{logit} [\mathbb{P}(Y_A = 1 | \Delta s(\mathbf{y}), \boldsymbol{\theta})] = \theta_0 + \sum_{l=1}^p \Delta_A s_l(\mathbf{y}) \theta_l = \theta_0 + \sum_{l=1}^p x_l \theta_l,$$

where x_l denotes the change statistics $\Delta_A s_l(\mathbf{y})$ which is considered as covariate in the logit model. Due to the induced conditional independence the likelihood for the sample results to

$$\mathcal{L}_{\mathcal{D}(n|2)}(\boldsymbol{\theta}) = \prod_{A \in \mathcal{D}(n|2)} \mathbb{P}(Y_A = 1 | \Delta_A s(\mathbf{y}), \boldsymbol{\theta}), \quad (4.7)$$

which is easily fitted using standard software for generalized linear models. Note that (4.7) is the true likelihood for the conditional subsample so that consistent estimates and their variance estimates are easily available. This means, by taking the subsample of edge variables Y_A with $A \in \mathcal{D}(n|2)$ and conditioning in the remaining graph we circumvent numerical estimation problems and remain in the classical generalized linear model framework. It also implies that we can estimate $\boldsymbol{\theta}$ consistently (for n increasing) by maximizing $\mathcal{L}_{\mathcal{D}(n|2)}(\boldsymbol{\theta})$.

Apparently, we may draw different samples of edges leading to different estimates. This means using different decomposition sets $\mathcal{D}(n|2)$ leads to different estimates. This leaves us with the question how to combine the different estimates. We may either draw $\mathcal{D}(n|2)$ randomly or make use of a combinatorial approach to cover the entire network \mathbf{Y} . Let

	1	2	3	4
1	0	1	2	3
2	1	0	3	2
3	2	3	0	1
4	3	2	1	0

Figure 4.2 *Symmetric Latin Square with unique diagonal.*

therefore

$$\mathcal{P} = \{\mathcal{D}_k(n|2), k = 1, \dots, n-1\}$$

be a sequence of sets $\mathcal{D}_k(n|2)$ such that each index pair is exactly in one single set $\mathcal{D}_k(n|2)$. That is for Y_{ij} there exists exactly one set $\mathcal{D}_k(n|2) \in \mathcal{P}$ with $(i, j) \in \mathcal{D}_k(n|2)$. The $n-1$ sets $\mathcal{D}_k(n|2)$ in \mathcal{P} can be constructed using a symmetric Latin Square with a unique diagonal (see, e.g., Andersen and Hilton, 1980).¹ For instance for $n = 4$ nodes Figure 4.2 shows a symmetric Latin Square. As we are focusing on undirected networks, where the corresponding adjacency matrix is symmetric, we use only the upper diagonal of the Latin Square. We may take the entries in the Latin Square as the sample number. For instance, $\mathcal{D}_1(n|2)$ results by taking the pairs with entries 1 in the upper triangle from the corresponding network adjacency matrix, i.e. $(1, 2), (3, 4)$ and condition on the remaining variables. Accordingly we proceed for entries 2 and 3 in the Latin Square. We denote with $\hat{\theta}_{<k>}$ the resulting estimate from sequence set $\mathcal{D}_k(n|2)$. Note that each estimate $\hat{\theta}_{<k>}$ is consistent but they are not mutually independent. With $Y_{<k>} = \{Y_{ij} : (i, j) \in \mathcal{D}_k(n|2)\}$ we easily get with the asymptotic properties of Maximum Likelihood estimates as $n \rightarrow \infty$ that

$$\mathbb{E}(\hat{\theta}_{<k>}) = \mathbb{E}_{\mathbf{Y} \setminus Y_{<k>}} \left(\mathbb{E}_{Y_{<k>}} \left(\hat{\theta}_{<k>} | \mathbf{Y} \setminus Y_{<k>} \right) \right) \rightarrow \theta.$$

¹ A description of a possible algorithm for the construction of such a symmetric Latin square with a unique diagonal is available, e.g., from Bogomolny (2016).

Moreover

$$\begin{aligned}\text{Var}(\hat{\theta}_{<k>}) &= \mathbb{E}_{\mathbf{Y} \setminus Y_{<k>}} \left(\text{Var}_{Y_{<k>}}(\hat{\theta}_{<k>} | \mathbf{Y} \setminus Y_{<k>}) \right) \\ &\quad + \text{Var}_{\mathbf{Y} \setminus Y_{<k>}} \left(\mathbb{E}_{Y_{<k>}}(\hat{\theta}_{<k>} | \mathbf{Y} \setminus Y_{<k>}) \right) \\ &\rightarrow \mathbb{E}_{\mathbf{Y} \setminus Y_{<k>}} \left(F_{<k>}^{-1}(\theta_{<k>}) \right),\end{aligned}$$

where $F_{<k>}(\theta)$ denotes the (conditional) Fisher matrix corresponding to the likelihood function (4.7). Apparently $F_{<k>}^{-1}(\theta_{<k>})$ is an unbiased estimate for $\mathbb{E}_{\mathbf{Y} \setminus Y_{<k>}} \left(F_{<k>}^{-1}(\theta_{<k>}) \right)$. Note that $F_{<k>}^{-1}(\hat{\theta}_{<k>})$ can be obtained with any software package for fitting logistic regression models. Hence an estimate for the variance is readily available.

4.3 Non-parametric Exponential Random Graph Models

4.3.1 Spline-Based Model

We have shown how an appropriate sample of the network allows for simple estimation of the parameters. Apparently this is a recommendable approach only if n , the number of nodes, is large. In this case, however, ERGMs become unstable if the change statistics increase linearly in n . As shown in Schweinberger (2011) this holds for almost all basic models with $\theta_l \neq 0$ for $l > 0$. In other words, even though we are able to estimate the parameters as described before, the resulting network will be either full or empty as n is becoming large. Stability is achieved if the network statistics are of order $O_p(n)$. One intention is therefore to modify the statistics in the model such that they become stable. This is done with non-parametric components so that the change statistics have a bounded influence. To do so we make use of the non-parametric model (4.3) where we additionally postulate that the smooth functions $m_l(\cdot)$ are monotone and bounded.

To estimate functions $m_l(\cdot)$ we make use of penalized spline smoothing as discussed in detail in Ruppert et al. (2003), and Ruppert et al. (2009), see also Kauermann et al. (2009). The general idea is as follows. First, one replaces the unknown smooth function $m_l(\cdot)$ by a spline basis which is flexible (i.e. high dimensional) enough to capture the underlying true functional relation. As a second step a penalty or regularization is imposed on the unknown spline coefficients leading to a smooth and numerically stable fit. The third step is to calibrate/estimate the amount of penalization, which is controlled by a smoothing parameter. The original idea goes back to O'Sullivan (1986) and was made popular by the seminal paper of Eilers and Marx (1996). We make use of the idea here, but amend it towards the specific problem of non-stability occurring in large networks. As first step

we choose a basis $\mathbf{B}(x) = (B_1(x), \dots, B_K(x))^t$ where $x \in \mathbb{R}^+$ and the basis components $B_q(x)$, for $q = 1, \dots, K$, fulfil the following three properties:

- 1) $B_q(0) \equiv 0$,
- 2) $B_q(x)$ is monotone, and
- 3) $B_q(x)$ is bounded for $x \rightarrow \infty$.

A convenient choice are distribution functions on \mathbb{R}^+ . Here we employ the exponential distribution and set

$$B_q(x) = 1 - \exp(-\gamma_q x), \quad (4.8)$$

where γ_q are fixed scaling parameters. The set $\{\gamma_1, \dots, \gamma_K\}$ covers a wide range of possible shapes as visualised in Figure 4.3. We now replace the unknown function $m_l(\cdot)$ in model (4.3) by the spline representation

$$m_l(x_l) = \mathbf{B}(x_l)^t \mathbf{u}_l, \quad (4.9)$$

with $\mathbf{B}(x) = (B_1(x), \dots, B_K(x))^t$ and $\mathbf{u}_l = (u_{l1}, \dots, u_{lK})^t$ as the coefficient vector. Note that as long as the coefficients of \mathbf{u}_l are finite we have constructed a bounded and hence stable network statistics. Apparently we need additional constraints on \mathbf{u}_l in order to guarantee monotonicity. This implies for monotonically increasing functions that

$$\mathbf{B}'(x)^t \mathbf{u}_l \geq 0, \quad (4.10)$$

where $\mathbf{B}'(x) = (\gamma_1 \exp(-\gamma_1 x), \dots, \gamma_K \exp(-\gamma_K x))^t$. This is a linear constraint on the parameters, which for estimation is easily accommodated by quadratic programming. For monotonically decreasing functions we use almost the same constraint. For practical purposes we select the cutpoints of neighbouring basis functions ξ_r with $\gamma_{r+1} \exp(-\gamma_{r+1} \xi_r) = \gamma_r \exp(-\gamma_r \xi_r)$ and set the constraints to $\mathbf{B}'(\xi_r)^t \mathbf{u}_l \geq 0$ for monotonically increasing functions (or to $\mathbf{B}'(\xi_r)^t \mathbf{u}_l \leq 0$ for monotonically decreasing functions), for $r = 1, \dots, K-1$. Our experiences show a stable behaviour with this setting.

4.3.2 Penalized Estimation

The second step is now to impose a penalty on the spline coefficients in order to achieve smoothness and numerical stability. For a sample of the network as proposed in the previous section, let $\ell(\theta_0, \mathbf{u})$ be the log-likelihood resulting from model (4.3) in combination with (4.9), where $\mathbf{u} = (\mathbf{u}_1^t, \dots, \mathbf{u}_p^t)^t$. For notational simplicity we omit the sampling index in this subsection. We emphasize however that the likelihood and hence its estimate do depend

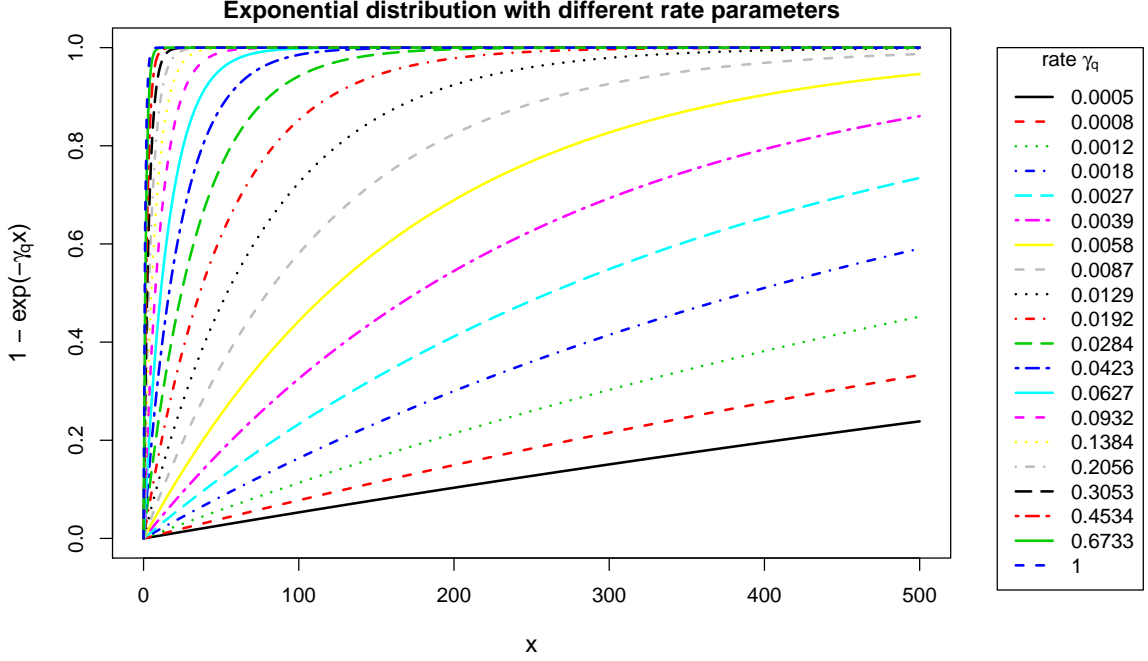


Figure 4.3 Visualisation of the cumulative distribution function of the exponential distribution with different rate parameters γ_q as example of possible basis functions $B_q(x)$.

on the particular sample of the network. Bear in mind that \mathbf{u} is high dimensional, so that (ML) estimates are unstable and the resulting fits $\mathbf{B}(x)\hat{\mathbf{u}}_l$ would be wiggled. We therefore apply a ridge penalty leading to the penalized log-likelihood

$$\ell_p(\theta_0, \mathbf{u}, \boldsymbol{\lambda}) = \ell(\theta_0, \mathbf{u}) - \frac{1}{2} \sum_{l=1}^p \lambda_l \mathbf{u}_l^t \mathbf{u}_l, \quad (4.11)$$

where $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_p)^t$ are the penalty parameters. Apparently setting $\lambda_l \rightarrow \infty$ leads to $m_l(\cdot) \equiv 0$ while $\lambda_l \rightarrow 0$ gives an unpenalized fit. It remains therefore to choose $\boldsymbol{\lambda}$ data driven balancing goodness of fit ($\boldsymbol{\lambda} \rightarrow 0$) and parsimony of the model (minimal for $\boldsymbol{\lambda} \rightarrow \infty$). These steps can be carried out with classical cross-validation (see, e.g., Eilers and Marx, 1996) or in a more sophisticated way by comprehending the penalty as normal prior. In this case we follow a Bayesian view and assume $\mathbf{u}_l \sim N(0, \lambda_l^{-1} I_K)$ with I_K as K dimensional unit matrix. Then λ_l is the reciprocal of the a priori variance of \mathbf{u}_l . The connection between penalized estimation and its Bayesian view by imposing normal priors is extensively motivated and discussed in Ruppert et al. (2003). In fact the approach led to a real breakthrough in smooth functional estimation as mirrored in the survey article by Ruppert et al. (2009). Note, that the Bayesian approach in our setting here leads to a generalized linear mixed model which is extensively discussed, e.g., in Breslow and

Clayton (1993), see also McCulloch et al. (2008). In particular, assuming a normal prior for coefficient vector \mathbf{u}_l we may consider the penalty λ_l as parameter which needs to be estimated. To do so we make use of the procedure of Schall (1991) leading to the following formulae. With $F(\theta_0, \mathbf{u})$ we denote the Fisher matrix of the conditional model (4.2). We define the Fisher matrix in the penalized likelihood (4.11) as

$$\mathbf{F}(\theta_0, \mathbf{u}, \boldsymbol{\lambda}) = \mathbf{F}(\theta_0, \mathbf{u}) + \text{diag}(0, \lambda_1 I_K, \dots, \lambda_p I_K),$$

where $\text{diag}(\cdot)$ denotes a block diagonal matrix with the arguments as blocks. The part of the Fisher matrix belonging to \mathbf{u} is then

$$\widetilde{\mathbf{F}}(\mathbf{u}, \boldsymbol{\lambda}) = \widetilde{\mathbf{F}}(\mathbf{u}) + \text{diag}(\lambda_1 I_K, \dots, \lambda_p I_K),$$

with $\widetilde{\mathbf{F}}(\mathbf{u})$ denoting the part of the Fisher matrix from the conditional model (4.2) belonging to \mathbf{u} . Following Schall (1991) we can now estimate λ_l^{-1} (iteratively) through

$$\widehat{\lambda}_l^{-1} = \frac{\mathbf{u}_l^t \mathbf{u}_l}{\text{df}(\lambda_l)}, \quad (4.12)$$

where

$$\text{df}(\lambda_l) = \text{tr} \left\{ \left[\widetilde{\mathbf{F}}^{-1}(\mathbf{u}, \boldsymbol{\lambda}) \widetilde{\mathbf{F}}(\mathbf{u}, 0) \right]_l \right\},$$

and subscript l means that we take only the submatrix matching to component \mathbf{u}_l . See Kauermann (2005), or Krivobokova and Kauermann (2007) for a derivation of the estimate. Finally, the monotonicity constraint (4.10) is taken into account by quadratic programming which is available in R using the package `quadprog` (Turlach and Weingessel, 2013). The following algorithm describes the iterative procedure.

Algorithm 1: Fit non-parametric ERGM, i.e. estimate $\boldsymbol{\beta} = (\theta_0, \mathbf{u}^t)^t$ and $\boldsymbol{\lambda}$.

Preparation: Fit Standard GLM for

$$\text{logit} \left[\mathbb{P}(Y_{ij} = 1 | \mathbf{Y} \setminus Y_{ij}; \boldsymbol{\theta}) \right] = \theta_0 + \sum_{l=1}^p \Delta_{ij} s_l(\mathbf{y}) \theta_l$$

to determine effect directions. The smooth effect $m_l(\cdot)$ is constrained to

- (a) a monotonically increasing function if $\widehat{\theta}_l \geq 0$, and
- (b) a monotonically decreasing function if $\widehat{\theta}_l < 0$.

Matrix \mathbf{A} is set up using the resulting monotonicity constraints according to (4.10).

Instead of maximizing $\ell_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) = \ell_p(\boldsymbol{\beta}, \boldsymbol{\lambda})$ directly under the constraints from (4.10), we use a Taylor expansion of

$$\ell_p(\boldsymbol{\beta}, \boldsymbol{\lambda}) - \ell_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda}) \approx \mathbf{s}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda})^t (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}) + \frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)})^t \mathbf{H}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda}) (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)}),$$

where $\mathbf{s}_p(\cdot)$ denotes the penalized score function and \mathbf{H}_p the penalized Hessian.

Initiate starting values $\boldsymbol{\beta}^{(0)}$, $\boldsymbol{\lambda}^{(0)}$, $t = 0$, $s = 0$.

Step 1: Use current value $\boldsymbol{\lambda}^{(s)}$ and iterate until convergence or until max. no. of iterations t_{\max} is reached:

- (i) Solve $\min_{\mathbf{b}} (-\mathbf{d}^t \mathbf{b} + 1/2 \mathbf{b}^t \mathbf{D} \mathbf{b})$ for $\mathbf{b} = (\boldsymbol{\beta} - \boldsymbol{\beta}^{(t)})$ with constraint $\mathbf{A}^t \mathbf{b} \geq \mathbf{b}_0$, where $\mathbf{d} = \mathbf{s}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda}^{(s)})$, $\mathbf{D} = -\mathbf{H}_p(\boldsymbol{\beta}^{(t)}, \boldsymbol{\lambda})$ and $\mathbf{b}_0 = -\mathbf{A}^t \boldsymbol{\beta}^{(t)}$.
- (ii) Update $\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \mathbf{b}$.
- (iii) Set $t = t + 1$.

Step 2: As long as maximum no. of iterations s_{\max} or convergence is not reached:

- (i) Use current value $\boldsymbol{\beta}^{(t)} = (\theta_0^{(t)}, (\mathbf{u}^{(t)})^t)^t$ and update to $\boldsymbol{\lambda}^{(s+1)}$ element-wise according to equation (4.12):

$$\hat{\lambda}_l^{(s+1)} = \frac{\text{tr} \left\{ \left[\widetilde{\mathbf{F}}^{-1}(\mathbf{u}^{(t)}, \boldsymbol{\lambda}^{(s)}) \widetilde{\mathbf{F}}(\mathbf{u}^{(t)}, 0) \right]_l \right\}}{(\mathbf{u}_l^{(t)})^t \mathbf{u}_l^{(t)}}, \quad \text{for } l = 1, \dots, p.$$

- (ii) Set $s = s + 1$.
- (iii) Set $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}^{(t)}$ and $t = 0$ and start again with *Step 1*.

Additional Steps:

When during fitting one of the penalty parameter λ_l tends to infinity (in *Step 2* (i)) we set the corresponding smooth effect $m_l(\cdot)$ to zero, i.e. $\widehat{m}_l(\cdot) \equiv 0$, and estimate the remaining components in the model with the above procedure.

If in a later iteration than the first iteration solving *Step 1* (i) fails (e.g., due to numeric problems), we take the current values $\boldsymbol{\beta}^{(t)}$ and check whether all of the estimated effects $\widehat{m}_l(\cdot)^{(t)}$, $l = 1, \dots, p$ exceed a specified threshold in absolute value.

If not, i.e. at least one effect estimate is close to zero, we set the smallest smooth effect to zero and continue.

4.3.3 Combining the Sample Estimates

Let us now bear in mind that the estimation described in the previous subsection holds for one sample of the network. For each sample we obtain an estimate $\widehat{m}_{l<k>}(x_l) = \mathbf{B}(x_l)\widehat{\mathbf{u}}_{l<k>}$ with obvious definition for $\widehat{\mathbf{u}}_{l<k>}$. We now need to combine the sample estimates which are mutually dependent. One possible approach for combining the sample estimates would be to calculate a mean curve by just averaging the resulting parameter estimates over all samples. We follow a different path here originating in functional data analysis and compute a median curve. The median curve is more robust against outliers than the mean curve. We employ the methods developed by Sun et al. (2012) which are available in the R package `fda` (Ramsay et al., 2014). There is a huge number of options available for computing functional depth, ranking curves accordingly, and determining a median curve (see, e.g., López-Pintado and Romo, 2009, and Mosler and Polyakova, 2012). We decided to use the approach of Sun et al. (2012) because it is quite fast even for a large number of curves, which seems important in our case (we use the option "Both" from the `fbplot` function `fda`, which first takes two curves for determining a band, and then computes a modified band depth in order to break ties between curves). We compute a joint median curve by sticking all estimated effects together. Computing marginal median curves per effect would be possible as well.

4.4 Data Example

4.4.1 Linear Estimation through Subsampling

As data example we use the combined data from ten Facebook ego networks, which has originally been collected by McAuley and Leskovec (2012) and is available from the Stanford Large Network Dataset Collection (Leskovec and Krevl, 2014). Figure 4.4 shows a plot of the network graph. It is undirected and contains 88,234 edges (Facebook friendships) between 4,039 nodes (actors). This amounts to a network density of roughly 0.01.

We use data from the first 4,038 rows and columns of the network adjacency matrix² and obtain 4,037 sample subsets $\mathcal{D}_k(n|2)$. To each of these subsets we fit a standard logistic model with edge (as intercept), 2-star, and triangle effect. We exclude subsets from the analysis which contain less than three observations with $y_{ij} = 3$ (this affects 56 data subsets). Figure 4.5 shows pairwise scatterplots of the estimated coefficients. Extreme results with an estimated intercept $\widehat{\theta}_{\text{edge}} < -10$ (115 estimates) are excluded. The general impression for the shown results is that the estimated triangle effect $\widehat{\theta}_{\text{triangle}}$ is always positive. The estimated 2-star effect $\widehat{\theta}_{2\text{-star}}$ is closer to zero with some positive

² As the number of nodes in the network has to be even for construction of the Latin square.

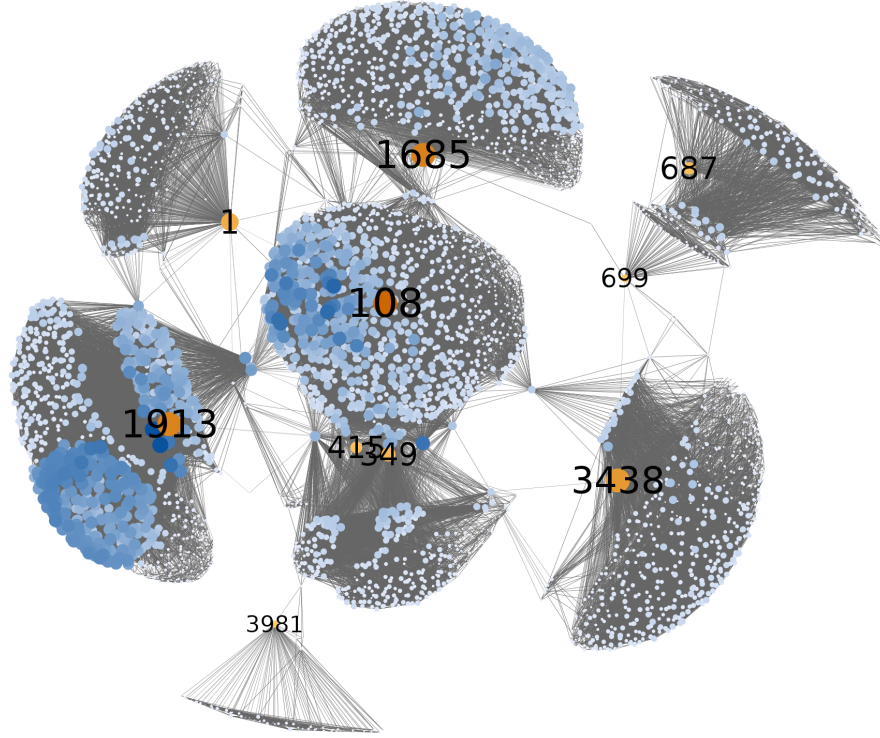


Figure 4.4 *Visualisation of the combined Facebook data. Colouring and size represent nodal degree (darker and bigger corresponds to higher degree). Darkness and thickness of links represents the no. of triangles, the link belongs to. The ten egos are highlighted with a label indicating the node number, and coloured in orange instead of blue. Generated using stress minimization layout in visone (Brandes and Wagner, 2004).*

and some negative values. There is some negative correlation between the two parameters. Table 4.1 displays a numerical summary of the results.

Apparently, in a network of this size we are faced with degeneracy using 2-stars and triangles as model statistics. We therefore do not put too much emphasis in the analysis of the parametric model but go forward to a non-parametric approach in the next subsection.

Table 4.1 *GLM (edge, 2-star, and triangle effect) results for the Facebook data. Extreme estimates with an estimated intercept $\hat{\theta}_{\text{edge}} < -10$ (115 estimates) are not considered.*

Parameter	mean est.	median est.	5% quantile	95% quantile
θ_{edge}	-5.436	-5.425	-7.373	-3.687
$\theta_{\text{2-star}}$	-0.012	-0.003	-0.054	0.006
θ_{triangle}	0.207	0.174	0.063	0.483

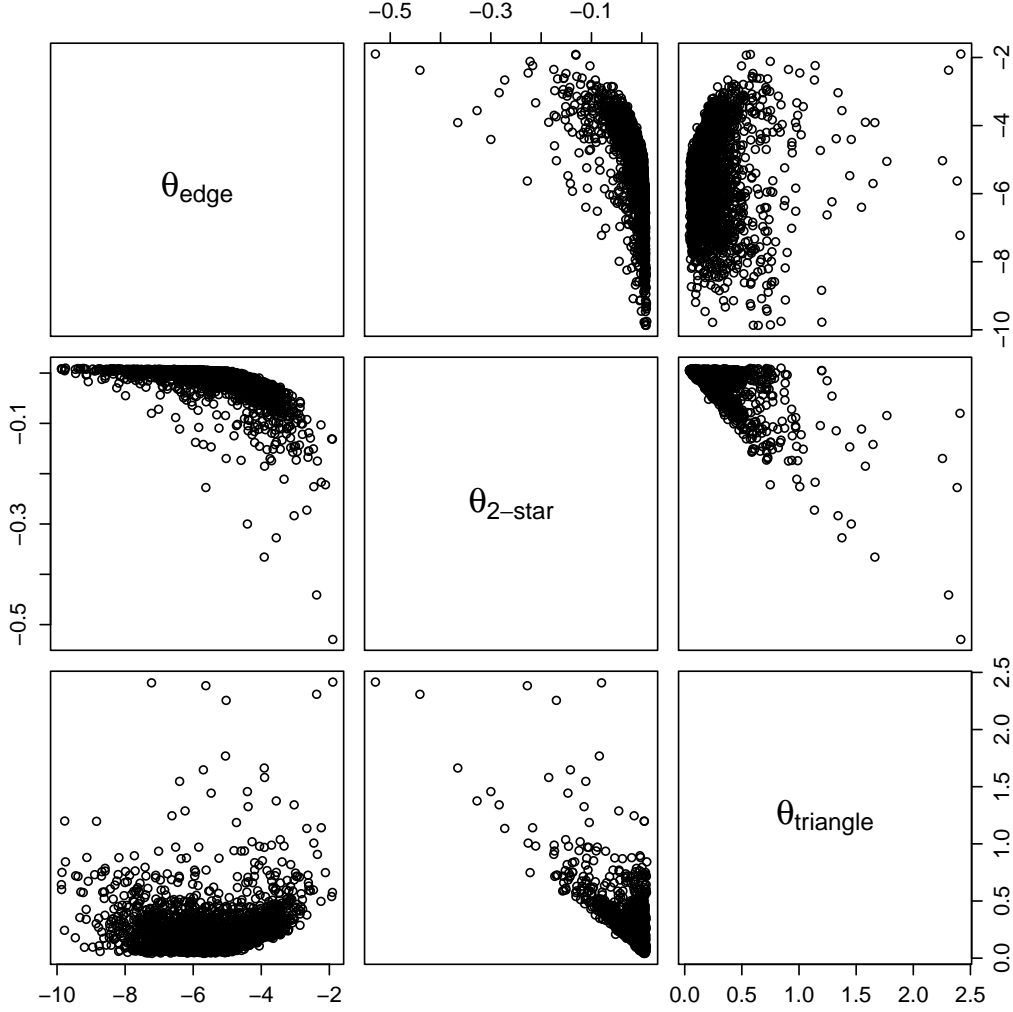


Figure 4.5 GLM (edge, 2-star, and triangle effect) results for the Facebook data. Extreme estimates with an estimated intercept $\hat{\theta}_{\text{edge}} < -10$ (115 estimates) are not shown.

4.4.2 Non-parametric Estimation through Subsampling

We stick to the Facebook data example and continue our analysis with a non-parametric Exponential Random Graph Model as described in Section 4.3. We fit a model containing the edge effect (as intercept), a smooth 2-star effect, and a smooth triangle effect.

We exclude subsets from the analysis which contain less than 10 observations with $y_{ij} = 1$, because otherwise we do not have enough information for a stable estimation of the smooth effects. This affects 577 out of 4,037 data subsets. We use 20 exponential distributions as basis functions for each smooth component, with parameters γ_q ranging between values of 0.0005 and 1 as displayed in Figure 4.3. The maximum number of iterations is 20,

which is not reached for any of the fits. The convergence criterion is set to $1e-12$ and for 83 samples the algorithm is aborted, which, e.g., may be caused by separability in these subsets and resulting in non-identifiability. As described in the algorithm in the previous section, the fitted model can simplify if the fitted λ_l goes to infinity. In this case the corresponding functional fit $\widehat{m}_l(\cdot)$ equals zero and the model is reduced. This implies that the fitting algorithm itself conducts a model selection. In addition, effects can be set to zero if numerical issues occur when solving the quadratic problem in *Step 1* of the algorithm and the current effect estimate $\widehat{m}_l(\cdot)$ is close to zero (we use a value of 0.005 for this criterion here). We therefore record the number of samples where the algorithm converges to a simplified model. Let the different models be labelled as follows:

$$\begin{aligned} M_1 &: \text{“2-star”} + \text{“triangle”} \\ M_2 &: \text{“triangle”} \\ M_3 &: \text{“2-star”} \\ M_4 &: \text{intercept only} \end{aligned}$$

The notation means that model M_2 , for instance, corresponds to a model where the non-parametric smooth 2-star effect is set to zero, while for model M_4 both, 2-star and triangle effect are set to zero. Table 4.2 summarises the results numerically and shows the number of samples for the converged models. There is a clear dominance for model M_2 with intercept and smooth non-parametric triangle effect only.

Figure 4.6 shows the resulting 3,377 estimates (subsets with convergence, or effect set to zero), containing mean (solid blue lines) and median estimates (dashed orange lines). As general impression we obtain a negative intercept, a positive triangle effect, and a 2-star

Table 4.2 *Numeric summary of the model fitting results for the Facebook data. The non-parametric ERGM contains an edge, a smooth 2-star, and a smooth triangle effect.*

Total no. of samples available:	4,037
No. of samples with no fit (less than 10 times $y = 1$ in sample):	577
Model M_1 :	181
Model M_2 :	3,189
Model M_3 :	5
Model M_4 :	2
No. of samples where max. no. iterations was reached:	0
No. of samples with other reason for non-convergence:	83

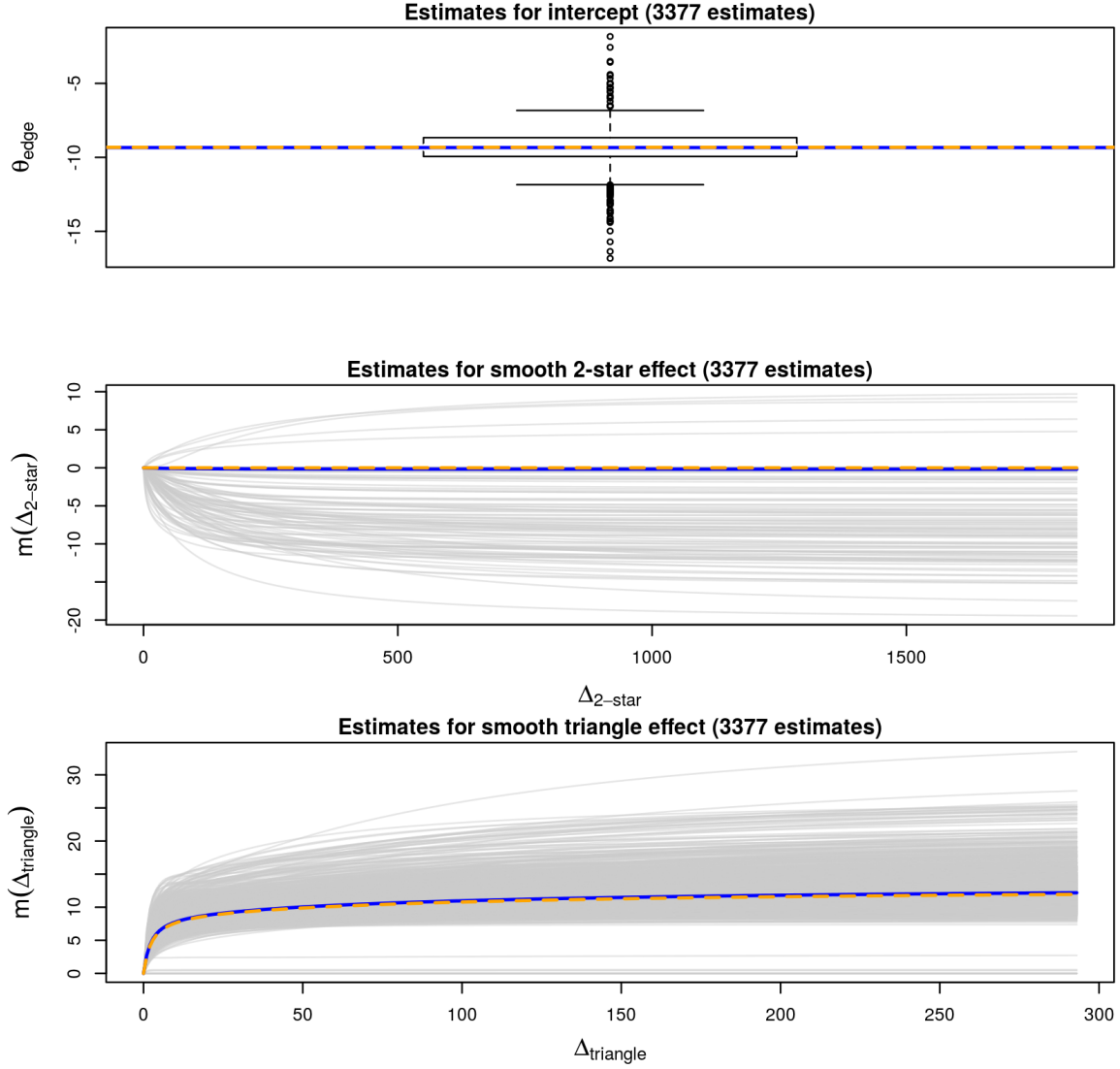


Figure 4.6 *Non-parametric ERGM (edge, smooth 2-star, and smooth triangle effect) results for the Facebook data. 3,377 estimates with convergence or effects set to zero are shown. The blue solid lines show the mean estimate; the orange dashed lines depict the median estimate.*

effect which is set to zero for most samples and fluctuates around zero in the remaining cases. The median curve for the 2-star effect is exactly zero.

To explore the validity of the model, we continue our analysis by computing Pearson residuals for all observations

$$e_{ij} = \frac{y_{ij} - \hat{\pi}_{ij}}{\sqrt{\hat{\pi}_{ij}(1 - \hat{\pi}_{ij})}}, \quad \text{for } i = 1, \dots, n, \quad j = 1, \dots, n, \quad i \neq j,$$

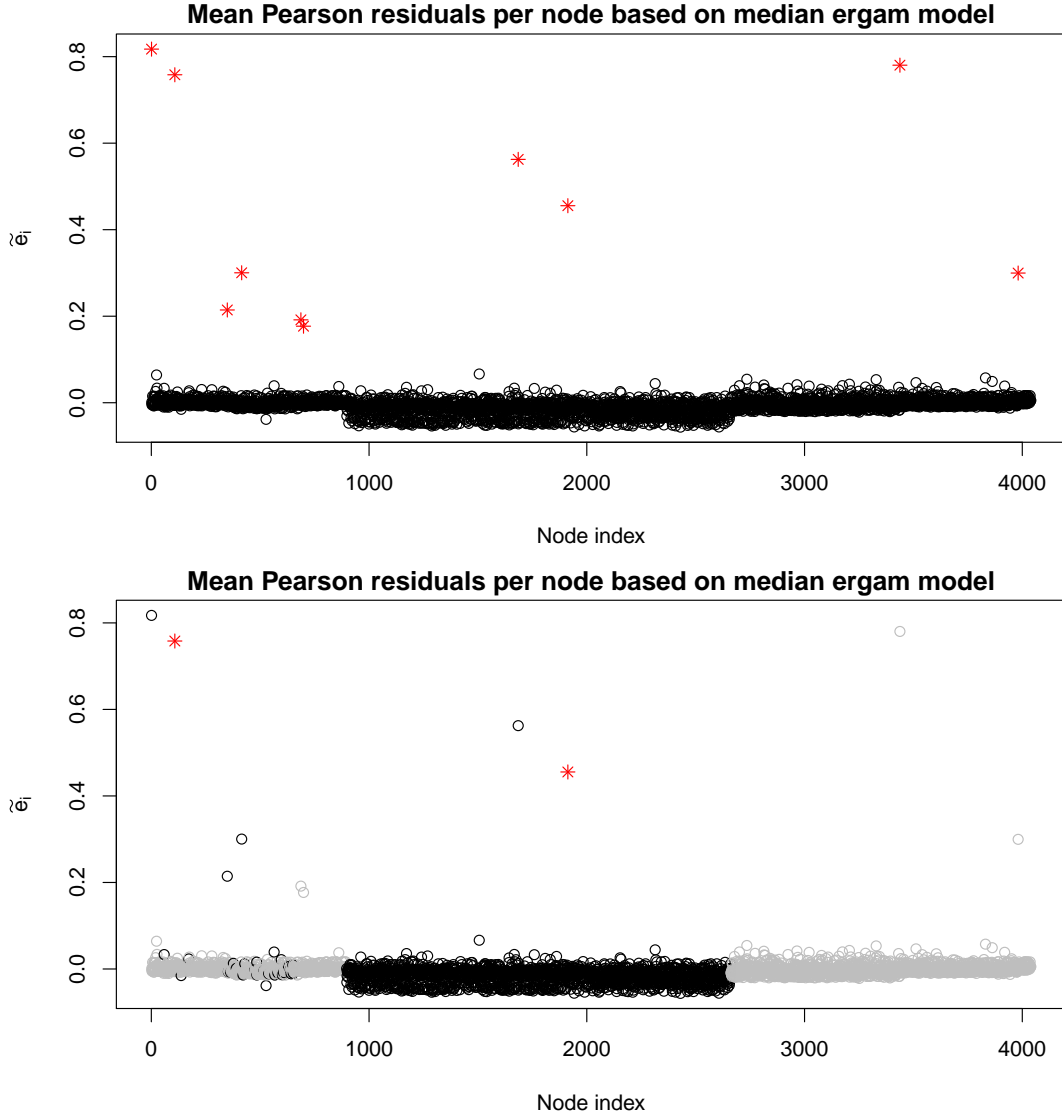


Figure 4.7 Node-specific average Pearson residuals from non-parametric ERGM for the Facebook data. Prediction for the residuals is based on the overall median model. The ten egos in the network are denoted with a red star in the upper plot. The lower plot highlights two ego-nets (for nodes 108 and 1,913). The two egos are denoted with a red star, the corresponding members of their ego-networks are black, the remaining ones of the whole dataset grey.

where $\hat{\pi}_{ij}$ is a prediction based on the obtained median (curve). As next step we calculate the average Pearson residual for each node through

$$\tilde{e}_i = \frac{1}{n-1} \sum_{j \neq i} e_{ij}, \quad \text{for } i = 1, \dots, n.$$

Figure 4.7 shows the resulting node-specific average Pearson residuals. Bear in mind that the residuals are not independent, as we are averaging over non-independent samples.

They should still have an expected value of zero. Figure 4.7 shows a clear structure. The nodes with large average residuals \tilde{e}_i are not surprising, as these are the ten egos from the network construction. In the upper plot all ten egos are depicted with a red star. They have more connections as one would expect from the model and therefore stick out. Moreover, some nodes have rather negative Pearson residuals and these nodes can be attributed to specific ego-nets. The lower plot in Figure 4.7 highlights two ego-nets (for nodes 108 and 1,913, these two egos are again depicted with a red star, the residuals belonging to the ego-nets are black, the remaining ones grey) and they account for almost all of these negative residuals. Our conclusion from this residual analysis is that for some parts of the network the overall model seems too simplistic, while for others the outcome appears to be reasonable.

We continue our analysis and look at the ego-nets of node 108 (contains a majority of nodes with negative average Pearson residual; consists of 1,045 nodes with 26,750 edges), and of node 1,685 (consists of 792 nodes with 14,025 edges) separately. The egos themselves are not part of the subnetworks as they are connected to every other vertex in the corresponding subnetwork (by construction). The setup for the fit is the same as before for the non-parametric ERGM. Figures 4.8, and 4.9 show the corresponding estimates. Table 4.3 summarises the results. When comparing the results to the ones for the whole dataset, the overall impression is similar, with a positive triangle effect, and a 2-star effect close to zero (or set to zero for most samples, and a zero median curve). The intercept values are smaller in absolute value, which is not surprising as we are analysing smaller networks (with a higher density).

Table 4.3 *Numeric summary of the model fitting results for ego-subnets of the Facebook data. The non-parametric ERGM contains an edge, a smooth 2-star, and a smooth triangle effect.*

Ego-net	108	1,685
Total no. of samples available:	1,043	791
No. of samples with no fit (less than 10 times $y = 1$ in sample):	132	65
Model M_1 :	0	10
Model M_2 :	911	716
Model M_3 :	0	0
Model M_4 :	0	0
No. of samples where max. no. iterations was reached:	0	0
No. of samples with other reason for non-convergence:	0	0

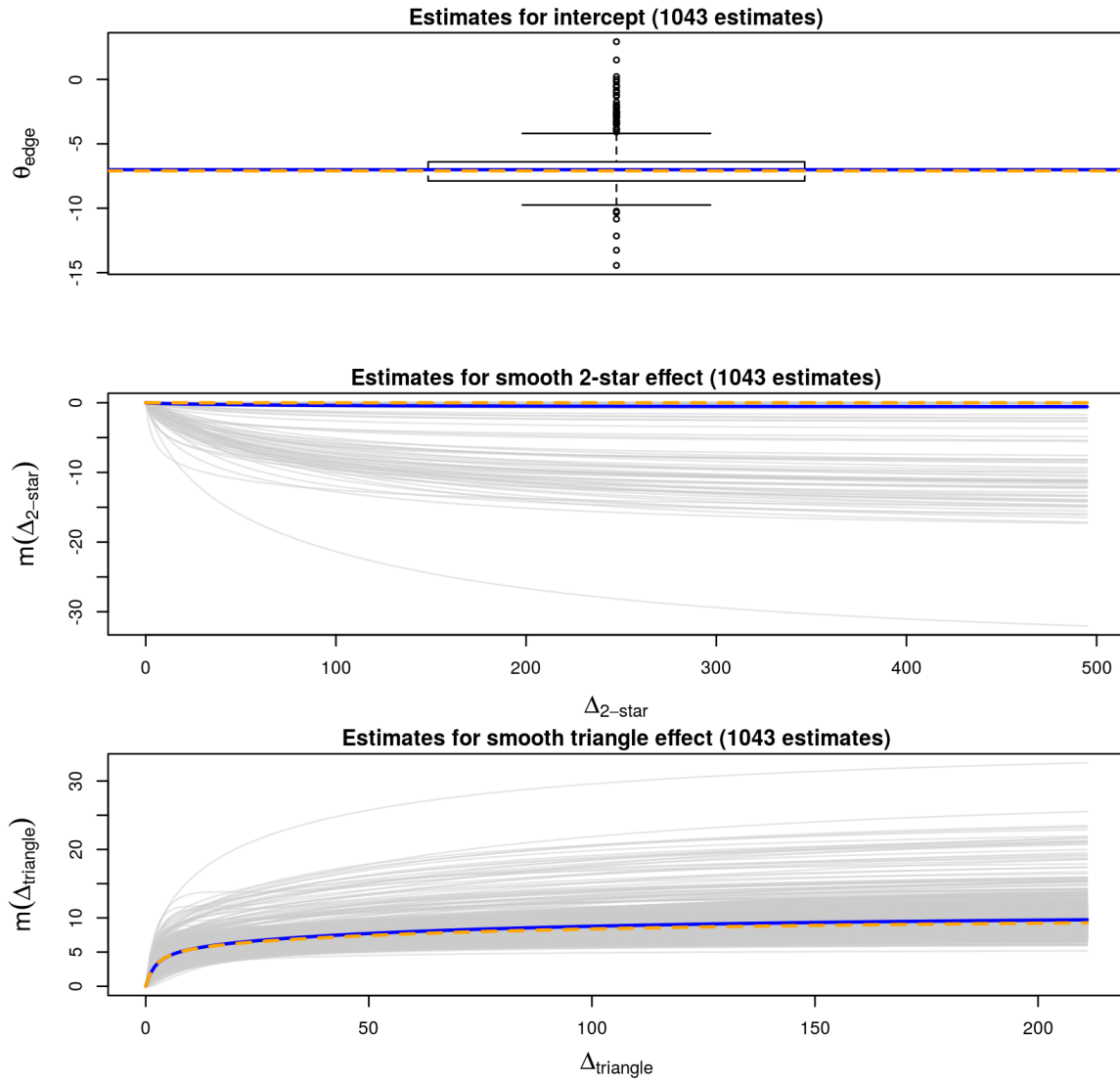


Figure 4.8 Non-parametric ERGM (edge, smooth 2-star, and smooth triangle effect) results for ego-net of node 108 from the Facebook data. 1,043 estimates with convergence or effects set to zero are shown. The blue solid lines show the mean estimate; the orange dashed lines depict the median estimate.

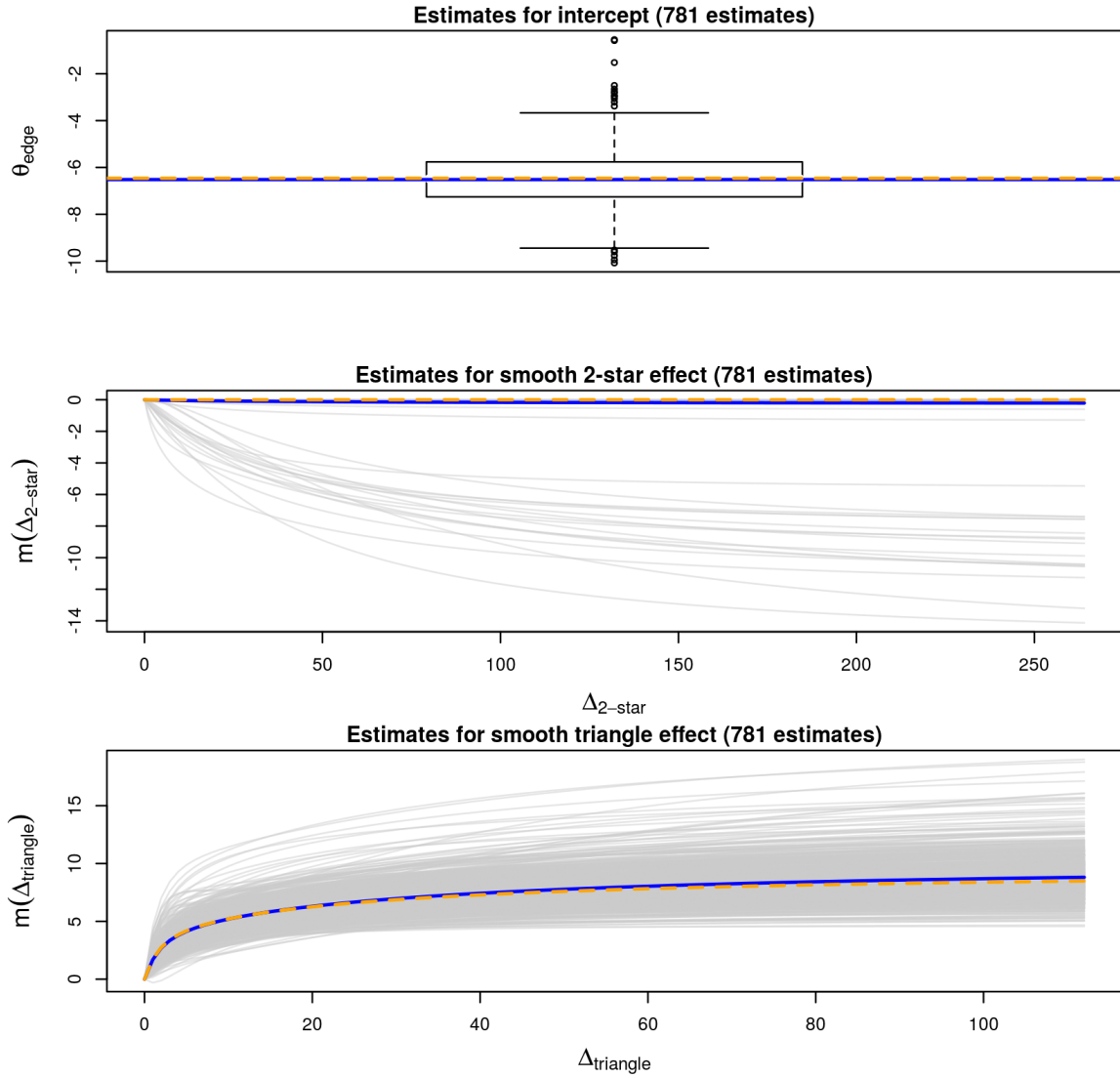


Figure 4.9 Non-parametric ERGM (edge, smooth 2-star, and smooth triangle effect) results for ego-net of node 1,685 from the Facebook data. 781 estimates with convergence, or effects set to zero are shown. The blue solid lines show the mean estimate; the orange dashed lines depict the median estimate.

Figure 4.10 shows the resulting node-specific average Pearson residuals for both ego-nets. The result looks more homogeneous than before, but for the ego-net of 108 we still see that there are nodes in the network with a rather negative average Pearson residual, i.e. they have fewer connections than the model would predict. This might be solved by extending the modelling approach and include node-specific or dyadic covariates into the model. This is of course easily possible in combination with the smooth effects but lies beyond the scope of this work.

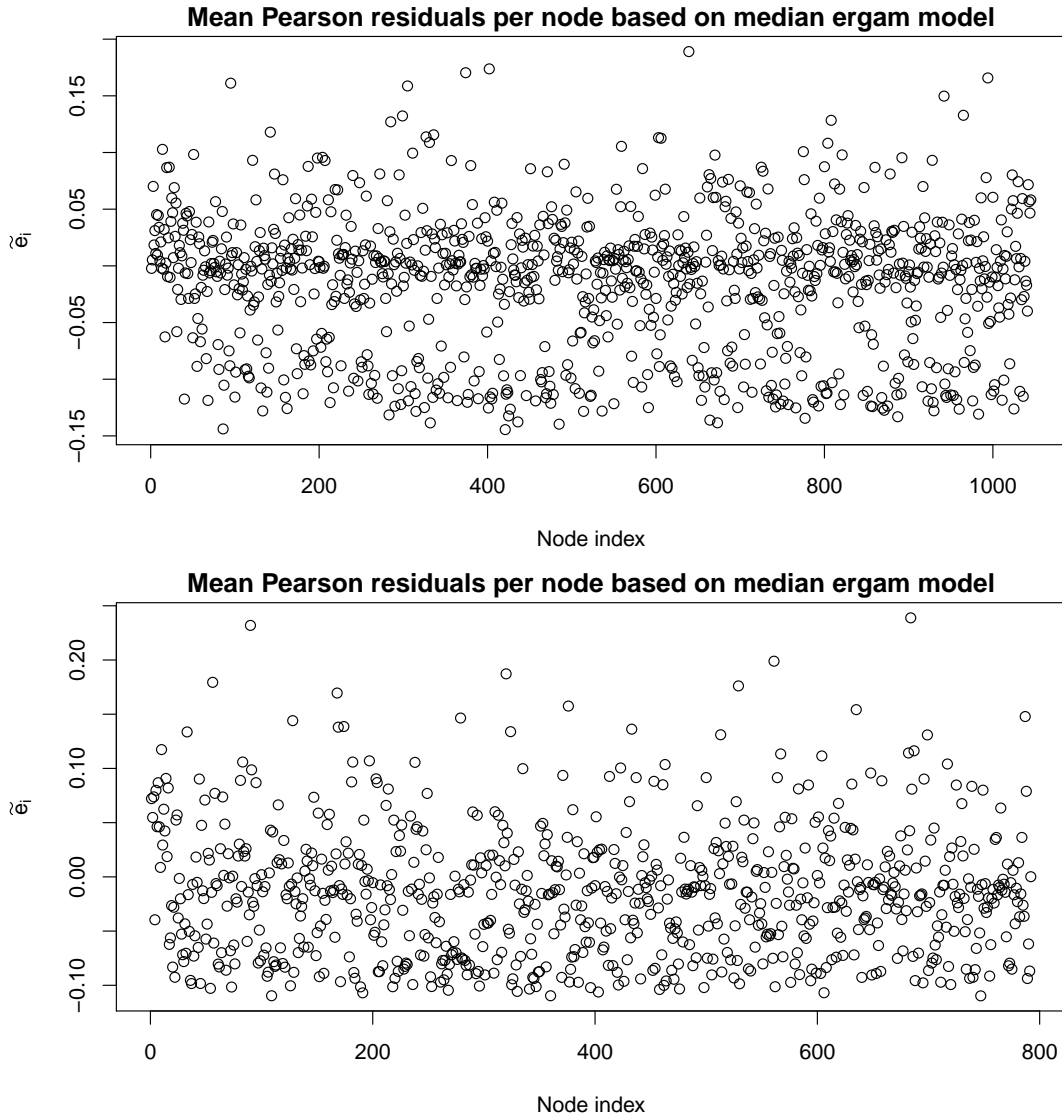


Figure 4.10 Node-specific average Pearson residuals from non-parametric ERGM for the ego-nets of node 108 (upper plot) and 1,685 (lower plot) from the Facebook data.

4.5 Discussion

We have shown that it is possible to make use of the Markov independence assumption in the context of Exponential Random Graph Models to obtain samples consisting of independent observations which allow to use standard generalized linear models (GLM) for model fitting. Extending this approach to generalized additive models (GAM) by adding smooth functional components in a non-parametric fashion enables us to gain flexibility while maintaining the simple interpretability of statistics like 2-stars and triangles. It circumvents the construction or use of more complex statistics like, e.g., geometrically weighted degree or edge-wise shared partners, which also stabilise the model fitting but are very difficult to interpret. In addition, the whole estimation procedure is quite fast (much faster than the standard MCMC based routines available for ERGMs) as we are using well established model fitting routines for GLMs and GAMs, and it can easily be run in parallel as the individual sample fits can be computed independently of each other. The computation for the Facebook data example was run in parallel on 20 cores (with 2.60 GHz) and took less than four minutes (including all data pre-processing and storing the results on disk).

To employ the described models the network needs to be big enough (otherwise the resulting samples are too small), and what can be more problematic, the network has to be dense enough as otherwise we obtain samples consisting only of observations with $y_{ij} = 0$. The latter is a general problem in real-world networks, as it is well-known that with increasing network size n the density tends to become smaller and smaller. The proposed modelling strategy therefore clearly has some caveats. Also, it is difficult to give a general advice on how many actors are needed for our method to work. When using the GLM approach on each subsample of size $\frac{n}{2}$ a smaller number of observations is reasonable than for the non-parametric GAM approach. In the data example we have presented in the previous section, the sample size itself is not an issue with 2,019 observations per subsample for the whole network, and 396 or 522, respectively, for the ego-nets, where we fit models with two smooth functional components plus an intercept term. Still, we had some problems with obtaining samples with enough $y_{ij} = 1$ observations per sample. Of course this issue becomes more severe when the network density (which is 0.01 for the complete Facebook data, and therefore quite high for a network of this size) goes down.

Another problem which is apparent from the residual analysis in Figures 4.7, and 4.10 is that the residuals are quite low in absolute value. This is a sign of underdispersion in the underlying binomial models and can be explained by zero-inflation, i.e. we have more zeros in the data than we would expect under the model. This result is not surprising, again due to the low density in large networks, where Exponential Random Graph Models tend to be problematic in general. There are approaches going into the direction of assuming local dependence structures, whereas the standard ERGM assumes a global rather strict dependence structure and is therefore probably unrealistic especially in the context of

large networks. Schweinberger and Handcock (2015) use hierarchical ERGMs, see also the corresponding R package `hergm` (Schweinberger et al., 2015), where the neighbourhood structure can be taken into account if it is known, or estimated as a latent construct using a Bayesian approach. The later is computationally very problematic and rather time consuming or even infeasible for large networks. Another possible solution to handle the zero-inflation using our subsampling approach would be the use of mixture models as available, e.g., in the R package `flexmix` (Leisch, 2004), and employ zero-inflation models (Grün and Leisch, 2008, Section 5.1) for binomial data to each sample. To us this appears to be a promising field for future research.

Acknowledgements

We gratefully acknowledge the help of Ulrik Brandes with the visualisation of the Facebook data example in Figure 4.4 using `visone` (version 2.16).

5 Further Ideas

The intention behind this chapter is to document further ideas for model extensions or improvements of the available routines, which have been considered during the development of this thesis.

Contributions

For the Bayesian approach, this is again joint work with Nial Friel (School of Mathematical Sciences and Insight: The National Centre for Data Analytics, University College Dublin, Ireland), Alberto Caimo (School of Mathematical Sciences, Dublin Institute of Technology, Ireland), and Göran Kauermann (Institut für Statistik, Ludwigs-Maximilians-Universität München, Germany). All authors were involved in the development and the discussion of the following ideas.

5.1 Speed-up Bayesian Approach

There are several ideas to speed up the Bayesian estimation routines presented in Chapter 3. As we are updating each of the nodal random effect parameters $\phi_i, i = 1, \dots, n$, in turn, an idea, which appeared promising at first, was to re-use the simulated network from the iteration step before. Instead of starting from scratch in the network simulation required for every parameter update, one could employ the already available network from the previous iteration as starting value of the required network simulation. The hope was to be able to use less iterations for the network simulation. Since only one nodal parameter ϕ_i is updated, the influence on a global network scale should be rather small. However, the whole procedure turned out to be much slower than the current implementation. At the moment, the `Bergm` routines rely on the very generic framework available in `ergm` and when simulating a network only the resulting network statistics are needed. Obtaining the whole adjacency matrix and handing it over to the simulation function as starting value in the next step is much more time consuming. Nevertheless, when implementing an estimation algorithm completely in C or C++, for instance, re-using the previously obtained network simulation could help to speed up the whole model fitting process.

Another idea was to use block instead of single-site updates of parameters in the Bayesian context. This is of course possible, but usually the acceptance rates go down, as more proposals are rejected. There is some room for improvement, by, e.g., tuning the proposal densities, the number of iterations, and so on, but as stated before, the Bayesian procedure based on the exchange algorithm remains infeasible for large networks with thousands of nodes.

For the Bayesian ERGM framework Bouranis et al. (2015) suggest the use of an approximate Bayesian Pseudo-Likelihood approach. The resulting posteriors are then calibrated to yield reasonable posterior estimates for the model. As the estimation is based on Bayesian Pseudo-Likelihood, it is in general applicable to large networks, at least from a computational point of view. Integrating our nodal random effects extension into this approach should be possible in general, but needs further investigation and yields one promising direction for future research.

5.2 Parallel Computing

As already mentioned, the numerical demand required for the presented classical estimation approaches and the Bayesian setting containing network simulations is quite challenging. The same is true for the simulation based goodness-of-fit procedures. Where possible, we have used parallelisation, for instance, for the path sampling involved in the Bayes factor computation in Section 3.3, or for the model computation on different subsamples in the previous chapter. In general, when several simulated networks are needed, it is possible to start several Markov chains in parallel, but depending on the size, this approach may need a lot of memory and does not help, if only a single simulated graph is needed. Maier et al. (2016) make use of the Markov structure discussed in Section 4.3 to run the network simulation itself in parallel using multiple cores. So instead of toggling only one tie, several of them are toggled at the same time. This approach allows to simulate large networks with thousands of nodes in the ERGM framework within minutes. Unfortunately parallelisation is not always possible, e.g., in our Bayesian estimation routine in Section 3.2, every iteration depends on the previous one. For approaches like the adaptive direction sampling of Caimo and Friel (2011), where several chains run simultaneously, parallelisation might be an option. Here the problem is that the chains need to communicate, which can again slow down things. Nevertheless, whenever possible, one should make use of parallel computation as it can dramatically reduce computation time – not only when dealing with statistical network analysis.

5.3 Pseudo-Likelihood Bootstrap

Desmarais and Cranmer (2012b) propose the use of Maximum Pseudo-Likelihood estimation (MPLE) in combination with a bootstrap approach to achieve reliable variance estimates. The basic idea is as follows: The desired model is fitted using the MPLE. Then, based on the obtained estimate, networks are simulated, and for each simulated network again the MPLE is computed. This allows to obtain an estimate for the variance of the MPLE and maybe even assess the bias of the estimator. Besides its drawbacks, as described in Section 2.3, Pseudo-Likelihood estimation is not computationally demanding (compared to the simulation based routines), and can be calculated using standard statistical software packages. It is in general computationally applicable to large networks consisting of thousands of nodes.

Unfortunately, if the specified model is unstable or (near) degenerate the approach does not solve these issues. Calculating the initial MPLE may be possible in most cases, but simulating networks based on this estimate may generate only full or empty graphs, for which even a reasonable MPLE cannot be computed. Besides this problem, even though the estimation can be carried out for large networks, simulating them can be rather time consuming. The just described parallelisation approach may be helpful in this context.

As the focus of this thesis was on stabilising Exponential Random Graph Models while maintaining interpretability, we did not concentrate on the Bootstrap idea for improving the Pseudo-Likelihood estimates. In addition, the subsampling approach from the previous chapter makes MPLE estimation for the whole network obsolete in this context. Still, when working with Maximum Pseudo Likelihood estimation, this Bootstrap-based approach can help to obtain reasonable estimates of uncertainty.

6 The Bottom Line

“ p^ [...] has indeed become the best statistical model in network science.”*

Stanley Wasserman, in a review of Lusher et al. (2013)

After dealing with statistical network analysis and Exponential Random Graph (or p^*) Models in particular for quite a while now, the résumé is ambivalent. Even though these models are – at first – very appealing from a statistical point of view due to their flexibility and exponential family type distribution, they have a lot of limitations, especially when it comes to large networks comprising thousands of nodes. This is not only due to the computational difficulties, such as time consuming algorithms or non-converging Markov chains, arising or aggravating with increasing network size. The main assumptions which induce a global dependency structure are often unrealistic in such situations. When analysing thousands of actors, the probability of becoming friends is different for neighbours (leaving aside what exactly defines the neighbourhood) than for completely strangers. This is also one of the reasons of the underdispersion or zero-inflation we see in the Facebook example in Chapter 4. It is common in many data examples as usually network density goes down with increasing network size. There are of course attempts as, for instance, the hierarchical ERGM of Schweinberger and Handcock (2015) to relax the global dependency assumption. This is achieved by defining a cluster-like structure, where within each cluster of nodes an ERGM is used and between clusters independence is assumed. However, if the clustering structure is completely unknown and therefore needs to be estimated as well, the models are again not applicable to large networks. In the context of the Stochastic Actor Oriented Models for dynamic network modelling similar issues arise. The current developments in this area try to solve this problem using so-called settings models where the social context, in which a network arises, is taken into account (see, e.g., Snijders et al., 2013; Lomi and Stadtfeld, 2014).

Specifying an Exponential Random Graph Model, that is deciding what statistics should be included into the model, is difficult. There is a variety of options available (see, e.g., Morris et al., 2008). Just throwing everything in and hoping for the best does not work. As the model fitting itself is often complicated and time consuming, the number of repeated model calculations is limited in a lot of cases. The researcher therefore needs to consider carefully, what might be potential driving forces of tie formation, what should

be included in the model, and what hypotheses should be tested – which is probably always a good idea to do before fitting any regression type model, not only in the network context. But even after sound construction, fitting the model may be problematic due to the aforementioned obstacles like degeneracy or non-convergence. Obtaining stable results by including geometrically weighted statistics comes at the price of not being able to easily interpret the results. We have seen in Chapter 3 that including nodal random effects accounts for unobserved heterogeneity of nodes in the network, while yielding easily interpretable results. Assuming nodes to be homogeneous (except for available nodal covariates), which is the standard assumption of ERGMs, is often unrealistic and can cause problems, such as instability of the results or wrong inferences. Again, for large networks the usage of the Bayesian approach presented in this thesis is currently not feasible. Including those nodal random effects into models fitted to the subsamples obtained using the strategy from Chapter 4 does not work either, as by construction each node only appears exactly once in each subsample as part of a single tie variable (otherwise we would not have independent observations). However, repeated measurements are required for the estimation of nodal random effects.

Including smooth functional components based on easily interpretable statistics stabilizes the models as well, especially for large networks. Still, this does not solve the global dependency issues. Employing mixture models to each subsample to cope with zero inflation in binary data, might be a solution. Mixture models are available, e.g., in the R package `flexmix` (Leisch, 2004, and Grün and Leisch, 2008, Section 5.1) and could be extended to combine local ERGMs (which can also contain smooth components) with a very low global probability that any two actors form a tie. In this way, one would integrate a notion of locality into the model.

Another difficulty arising for Exponential Random Graph Models is the comparison of competing models, especially if a formal comparison is desired and a sole visual inspection of goodness-of-fit plots is not sufficient. The presented Bayes factor computation in Chapter 3 allows for such a comparison of arbitrary Bayesian Exponential Random Graph Models. Model complexity is taken into account in this approach. We have shown that more complex models are not systematically preferred. Again, at the moment, this only works for small networks, where Bayesian ERGMs (with model fitting based on the exchange algorithm) are applicable.

Exponential Random Graph Models are nonetheless a great tool to combine structural network effects with nodal or dyadic covariates and can capture a lot of factors, such as structural, nodal, or dyadic effects, which (potentially) influence tie formation in a network. Triadic closure is probably one the most prominent ones of these features. The advantages and potentialities of ERGMs have been described extensively in the literature, see, e.g., Lusher et al. (2013). Unfortunately this convenient modelling strategy only applies to reasonably small networks up to a couple of hundreds of nodes. As explained before, the

models tend to become very problematic, not only in estimation, but even more in their underlying assumptions, when dealing with larger networks which are of increasing interest for researchers and industry. As more and more large datasets become available, which yield interesting research questions, the applicability of analytical methods to these high amounts of data is in focus – and we are not even dealing with real “Big Data” with millions of observations yet.

Whether the Exponential Random Graph framework can be extended to yield a reasonable approach for large network datasets remains questionable at the moment. Maybe one needs to get away from the wish to analyse the whole dataset with a single model. Identifying sub-groups / sub-networks in the data for which the required model assumptions hold is an option. Of course, due to the inconsistency under sampling of ERGMs, this limits the generalizability of the results to the greater population. Nevertheless, obtaining reasonable results from a model for a sub-network, where the assumptions hold, is currently in our view a better strategy than fitting a questionable model to the whole network, which produces suspicious and potentially misleading results.

Referring to Wassermann’s quote at the beginning of this chapter: Exponential Random Graph Models are for sure one of the best statistical modelling approaches currently available in network data analysis. Nonetheless, there is still a lot of room for improvement, especially – but not only – when it comes to large networks with more than a classroom full of actors.

Appendices

A Network Statistics

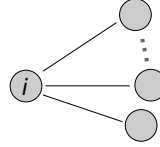
A.1 Some Examples with Notation and Formulae

Again, we assume that the network with observed adjacency matrix \mathbf{y} is undirected.

Degree

Degree of node i :

$$d_i = \sum_{j=1}^n y_{ij}$$



Degree of node i , ignoring the link to node j :

$$d_i^{-j} = \sum_{\substack{m=1 \\ m \neq j}}^n y_{im}$$

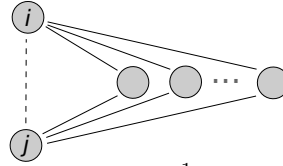
No. of nodes in \mathbf{y} with degree m :

$$D_m(\mathbf{y}) = \sum_{i=1}^{n-1} \mathbb{I}[d_i = m], \quad \text{where } \mathbb{I}[\cdot] \text{ denotes an indicator function}$$

Shared Partners

No. of shared partners of nodes i and j :

$$\text{sp}_{ij}(\mathbf{y}) = \sum_{m=1}^n y_{im} y_{mj}$$



No. of edges in \mathbf{y} with m shared partners:

$$\text{EP}_m(\mathbf{y}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n y_{ij} \mathbb{I}[\text{sp}_{ij} = m]$$

No. of dyads in \mathbf{y} with m shared partners:

$$\text{DP}_m(\mathbf{y}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \mathbb{I}[\text{sp}_{ij} = m]$$

Paths

No. of distinct paths of length r between nodes i and j :

$$[\mathbf{y}^r]_{ij}, \quad \text{where } \mathbf{y}^r \text{ denotes the } r\text{-th power of the adjacency matrix and } [\cdot]_{ij} \text{ is the entry in row } i \text{ and column } j$$

Table A.1 Illustrations and formulae for some network statistics with corresponding change statistics for which the Markov independence assumption holds, see Hunter (2007), and Goodreau (2007).



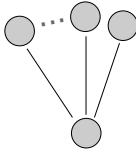
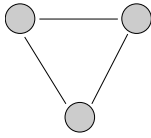
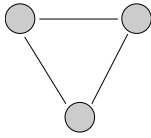
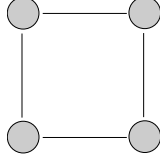
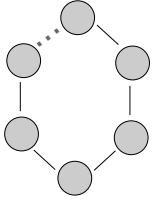
Term	Network Statistic $s(\mathbf{y})$	Change Statistic $s_{ij}(\mathbf{y}) = \Delta_{ij} s(\mathbf{y})$	Interpretation / Notes
Edges	 $\sum_{i=1}^{n-1} \sum_{j=i+1}^n y_{ij}$	1	Intercept
2-stars	 $\sum_{m=k}^{n-1} \binom{m}{k} D_m(\mathbf{y})$	$d_i^{-j} + d_j^{-i}$	Propensity of actors to form several ties (i.e. more than one)
k-stars	 $\sum_{m=k}^{n-1} \binom{m}{k} D_m(\mathbf{y}) = \sum_{i=1}^n \binom{d_i}{k}$	$\binom{d_i^{-j}}{k-1} + \binom{d_j^{-i}}{k-1}$	
GWD	 $e^{\theta_{\text{dec}}} \sum_{m=1}^{n-1} \left\{ 1 - \underbrace{(1 - e^{-\theta_{\text{dec}}})^m}_{=\varrho} \right\} D_m(\mathbf{y})$	$\varrho^{d_i^{-j}} + \varrho^{d_j^{-i}}$	Geometrically Weighted Degree
Triangles	 $\sum_{i=1}^{n-2} \sum_{j=i+1}^{n-1} \sum_{m=j+1}^n y_{ij} y_{jm} y_{mi} = \text{tr}(\mathbf{y}^3)/6$	$\text{sp}_{ij}(\mathbf{y}) = [\mathbf{y}^2]_{ij}$	No. of common friends of nodes i and j

Table A.2 Illustrations and formulae for some network statistics with corresponding change statistics for which the Markov independence assumption is violated, see Hunter (2007), and Goodreau (2007).

Term	Network Statistic $s(\mathbf{y})$	Change Statistic $s_{ij}(\mathbf{y}) = \Delta_{ij} s(\mathbf{y})$	Interpretation / Notes
GWESP	$e^{\theta_{\text{dec}}} \sum_{m=1}^{n-2} \left\{ 1 - \left(1 - e^{-\theta_{\text{dec}}} \right)^m \right\} \text{EP}_m(\mathbf{y})$	$s(\mathbf{y} \setminus y_{ij}, y_{ij} = 1) - s(\mathbf{y} \setminus y_{ij}, y_{ij} = 0)$	Geometrically Weighted Edge-wise Shared Partners
GWDSP	$e^{\theta_{\text{dec}}} \sum_{m=1}^{n-2} \left\{ 1 - \left(1 - e^{-\theta_{\text{dec}}} \right)^m \right\} \text{DP}_m(\mathbf{y})$	$s(\mathbf{y} \setminus y_{ij}, y_{ij} = 1) - s(\mathbf{y} \setminus y_{ij}, y_{ij} = 0)$	Geometrically Weighted Dyadwise Shared Partners
4-cycles	 $tr(\mathbf{y}^4)/8$	$[\mathbf{y}^3]_{ij}$	
k-cycles	 $tr(\mathbf{y}^k)/(2 \cdot k)$	$[\mathbf{y}^{k-1}]_{ij}$	

$tr(\cdot)$ denotes the trace of a matrix; θ_{dec} is an additional parameter defining the decay, which can either be fixed, or estimated (in Curved ERGMs).

A.2 Illustration of Markov Independence

This section contains a short illustration on why the Markov independence assumption is violated for the (geometrically weighted) edge-wise shared partner statistic, and why it holds for the (geometrically weighted) degree statistic. Markov independence means, that two edge variables Y_{ij} and Y_{kl} are conditionally independent, given the rest of the network, if they have no incident nodes, i.e. $i \neq j \neq k \neq l$.

We analyse the following network with four nodes (left figure). The induced Markov independence graph is shown in the figure on the right.

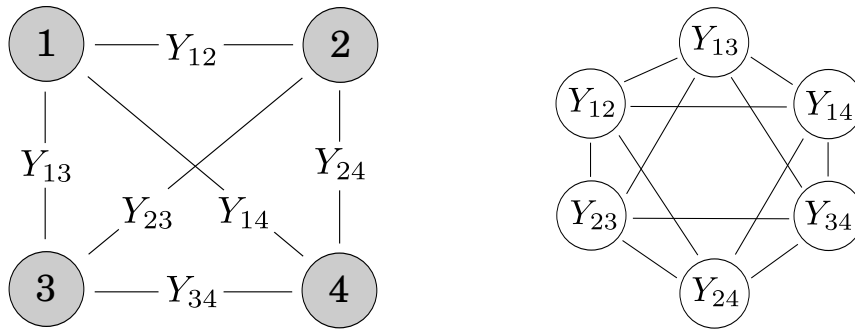
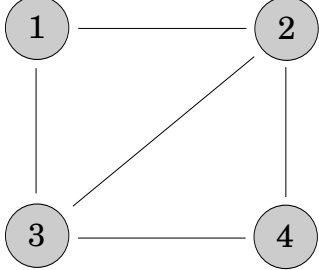
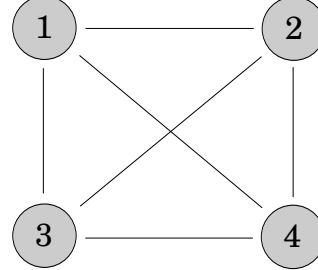


Figure A.1 Visualisation of the induced Markov independence graph (right) for a simple 4-node network (left).

If Markov independence holds, the variables Y_{14} and Y_{23} are conditionally independent, given the rest of the network. This implies that the status of Y_{23} does not alter the change statistic of Y_{14} . We therefore use two scenarios to illustrate the influence on the number of nodes with degree m , that is $D_m(\mathbf{y})$, for $m = 1, 2, 3$, and the number of edges with m shared partners, that is $EP_m(\mathbf{y})$, for $m = 1, 2$. The corresponding values are given in each cell.

	$Y_{14} = 0$	$Y_{14} = 1$
Scenario I: $Y_{23} = 0$	<p>no edge-wise shared partners</p> <p>$D_2(\mathbf{y}) = 4$</p>	<p>$EP_1(\mathbf{y}) = 4, EP_2(\mathbf{y}) = 1$</p> <p>$D_2(\mathbf{y}) = 2, D_3(\mathbf{y}) = 2$</p>

	$Y_{14} = 0$	$Y_{14} = 1$
Scenario II: $Y_{23} = 1$	 <p> $EP_1(\mathbf{y}) = 4, EP_2(\mathbf{y}) = 1$ $D_2(\mathbf{y}) = 2, D_3(\mathbf{y}) = 2$ </p>	 <p> $EP_2(\mathbf{y}) = 6$ $D_3(\mathbf{y}) = 4$ </p>

We use the general formula for the change statistic computation

$$s_{14}(\mathbf{y}) = \Delta_{14} s(\mathbf{y}) = s(\mathbf{y} \setminus y_{14}, y_{14} = 1) - s(\mathbf{y} \setminus y_{14}, y_{14} = 0).$$

In the two scenarios we obtain the following values for geometrically weighted edge-wise shared partners (GWESP) change:

$$\text{Scenario I: } \Delta_{14} s_{\text{GWESP}}(\mathbf{y}) = e^{\theta_{\text{dec}}} \left[\left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^1 \right\} \cdot 4 + \left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^2 \right\} \cdot 1 \right]$$

$$\text{Scenario II: } \Delta_{14} s_{\text{GWESP}}(\mathbf{y}) = e^{\theta_{\text{dec}}} \left[\left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^2 \right\} \cdot 5 - \left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^1 \right\} \cdot 4 \right]$$

Clearly the Markov assumption does not hold here, as the status of Y_{23} has an influence on the change statistic of Y_{14} and the values are not the same in both scenarios.

For geometrically weighted degree (GWD) the Markov independence assumption holds and we obtain the same change statistic in both scenarios:

$$\begin{aligned} \text{Scenario I/II: } \Delta_{14} s_{\text{GWD}}(\mathbf{y}) &= e^{\theta_{\text{dec}}} \left[\left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^3 \right\} \cdot 2 - \left\{ 1 - (1 - e^{-\theta_{\text{dec}}})^2 \right\} \cdot 2 \right] \\ &= 2 \cdot (1 - e^{-\theta_{\text{dec}}})^2 \end{aligned}$$

B R Code – Near-Degeneracy Illustration

R code for the illustration of the near-degeneracy problem in Figure 2.2 and the stable setting in Figure 2.3.

```
## Small simulation to illustrate near-degeneracy issue ##
library("ergm")

theta_edges <- rep.int(-2, times = 101)
theta_twostars <- seq(from = -1, to = 1, length.out = 101)

coefs <- cbind(theta_edges, theta_twostars)

n_nodes <- 30
n_edges_max <- n_nodes * (n_nodes - 1) / 2

nw <- network(n_nodes, directed = FALSE)

formula_sim <- nw ~ edges + kstar(2)
n_sim <- 50

control <- control.simulate.formula(MCMC.burnin = 5000,
                                     MCMC.interval = 4000)

ave_density <- function(coefs) {
  net_sim <- simulate(formula_sim,
                      coef = coefs,
                      nsim = n_sim, statonly = TRUE,
                      control = control)
  return(mean(net_sim[, 1] / n_edges_max))
}

set.seed(23)
resulting_density <- apply(coefs, MARGIN = 1, FUN = ave_density)
```

```
plot(theta_twostars, resulting_density, type = "l")

## Stable setting ##

theta_triangles <- - 3 * theta_twostars
coefs <- cbind(theta_edges, theta_twostars, theta_triangles)

formula_sim <- nw ~ edges + kstar(2) + triangles

set.seed(23)
resulting_density <- apply(coefs, MARGIN = 1, FUN = ave_density)

plot(theta_twostars, resulting_density, type = "l")
```

C Laplace Approximation

The likelihood in the mixed effects model marginalized over the random effects ϕ is

$$\begin{aligned}
 f(\mathbf{y}|\boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2) &= \int \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y}) + \boldsymbol{\phi}^t t(\mathbf{y})\}}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})} \cdot p(\boldsymbol{\phi}|\mu_\phi, \sigma_\phi^2) d\boldsymbol{\phi} \\
 &= \int \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y}) + \boldsymbol{\phi}^t t(\mathbf{y})\}}{\kappa(\boldsymbol{\theta}, \boldsymbol{\phi})} \cdot \frac{1}{(2\pi)^{\frac{n}{2}} |\sigma_\phi^2 I_n|^{\frac{1}{2}}} \\
 &\quad \cdot \exp\left\{-\frac{1}{2\sigma_\phi^2}(\boldsymbol{\phi} - \mu_\phi \mathbf{1}_n)^t (\boldsymbol{\phi} - \mu_\phi \mathbf{1}_n)\right\} d\boldsymbol{\phi} \\
 &= \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}}{(2\pi\sigma_\phi^2)^{\frac{n}{2}}} \cdot \\
 &\quad \cdot \int \exp\left\{\boldsymbol{\phi}^t t(\mathbf{y}) - \frac{1}{2\sigma_\phi^2}(\boldsymbol{\phi} - \mu_\phi \mathbf{1}_n)^t (\boldsymbol{\phi} - \mu_\phi \mathbf{1}_n) - \log(\kappa(\boldsymbol{\theta}, \boldsymbol{\phi}))\right\} d\boldsymbol{\phi}.
 \end{aligned} \tag{C.1}$$

The integral in equation (C.1) is approximated around the point $\hat{\boldsymbol{\phi}}$ using a Laplace type approximation (Severini, 2000, section 2.11)

$$\int \exp\{-h(\boldsymbol{\phi})\} d\boldsymbol{\phi} \approx \exp\{-h(\hat{\boldsymbol{\phi}})\} (2\pi)^{\frac{n}{2}} |\Sigma|^{-\frac{1}{2}}, \tag{C.2}$$

where

$$h(\boldsymbol{\phi}) = -\boldsymbol{\phi}^t t(\mathbf{y}) + \frac{1}{2\sigma_\phi^2}(\boldsymbol{\phi} - \mu_\phi \mathbf{1}_n)^t (\boldsymbol{\phi} - \mu_\phi \mathbf{1}_n) + \log(\kappa(\boldsymbol{\theta}, \boldsymbol{\phi}))$$

and

$$\begin{aligned}
 \Sigma &= \frac{\partial^2 h(\hat{\boldsymbol{\phi}})}{\partial \hat{\boldsymbol{\phi}} \partial \hat{\boldsymbol{\phi}}^t} \\
 &= \frac{1}{\sigma_\phi^2} I_n + \frac{\partial^2}{\partial \hat{\boldsymbol{\phi}} \partial \hat{\boldsymbol{\phi}}^t} \log(\kappa(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})) \\
 &= \frac{1}{\sigma_\phi^2} I_n + \text{Cov}(t(\mathbf{Y}), t(\mathbf{Y})^t | \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}).
 \end{aligned}$$

The matrix $\text{Cov}\left(t(\mathbf{Y}), t(\mathbf{Y})^t | \hat{\boldsymbol{\phi}}, \boldsymbol{\theta}\right)$ denotes the covariance matrix of the vector of degree statistics $t(\mathbf{Y})$ and can be estimated via simulated networks using the parameters $\hat{\boldsymbol{\phi}}$ and $\boldsymbol{\theta}$. These networks are drawn in the same way as the auxiliary networks needed for the exchange algorithm described in Section 3.2.

We assume that the posterior mode is close to the maximum likelihood estimator. The two are identical if the prior distributions are non-informative. This is not the case here, but we are assuming flat prior distributions and therefore the two should be reasonably close to each other. For reasons of simplicity, we use the posterior mean as value for $\hat{\boldsymbol{\phi}}$.

Combining equation (C.1) with equation (C.2) yields

$$f(\mathbf{y} | \boldsymbol{\theta}, \mu_\phi, \sigma_\phi^2) \approx \frac{\exp\{\boldsymbol{\theta}^t s(\mathbf{y})\}}{\kappa(\boldsymbol{\theta}, \hat{\boldsymbol{\phi}})} \hat{f}_{\text{Laplace}}(\mathbf{y} | \hat{\boldsymbol{\phi}}, \mu_\phi, \sigma_\phi^2), \quad (\text{C.3})$$

with

$$\hat{f}_{\text{Laplace}}(\mathbf{y} | \hat{\boldsymbol{\phi}}, \mu_\phi, \sigma_\phi^2) = \sigma_\phi^{-n} \exp \left\{ \hat{\boldsymbol{\phi}}^t t(\mathbf{y}) - \frac{1}{2\sigma_\phi^2} (\hat{\boldsymbol{\phi}} - \mu_\phi \mathbf{1}_n)^t (\hat{\boldsymbol{\phi}} - \mu_\phi \mathbf{1}_n) \right\} |\Sigma|^{-\frac{1}{2}}.$$

List of Figures

2.1	Examples of network sub-graphs.	9
2.2	Illustration of near-degeneracy issue (edge + 2-star effect).	16
2.3	Illustration of stable setting (edges + 2-star + triangle effect).	17
3.1	Overview: Bayesian model formulation for ERGM with nodal random effects.	27
3.2	Karate club graph with estimated nodal effect.	34
3.3	Karate club data: Model diagnostics (edge + triangle effect).	36
3.4	Karate club data: Model diagnostics (triangle + random effect).	36
3.5	Karate club data: Posterior densities (triangle vs. random effect).	37
3.6	Karate club data: Bayesian GOF diagnostics (edge + triangle effect).	38
3.7	Karate club data: Bayesian GOF diagnostics (triangle + random effect).	38
3.8	Karate club data: Model diagnostics (edge + GWESP effect).	40
3.9	Karate club data: Model diagnostics (GWESP + random effect).	41
3.10	Karate club data: Model diagnostics (random effect only).	41
3.11	Karate club data: Bayesian GOF diagnostics (edge + GWESP effect).	43
3.12	Karate club data: Bayesian GOF diagnostics (GWESP + random effect).	43
3.13	Karate club data: Bayesian GOF diagnostics (random effect only).	43
3.14	Kapferer graph with estimated nodal effect.	44
3.15	Kapferer data: Model diagnostics (edge + GWD + 2-star effect).	46
3.16	Kapferer data: Model diagnostics (random effect only).	46
3.17	Kapferer data: Bayesian GOF diagnostics (edge + GWD + 2-star effect).	47
3.18	Kapferer data: Bayesian GOF diagnostics (random effect only).	47
3.19	EP graph.	48
3.20	EP data: Model diagnostics (edge + triangle effect).	49
3.21	EP data: Model diagnostics (triangle + random effect).	49
3.22	Simulation: Resulting log Bayes factors for nested models.	51
3.23	Simulation: Resulting log Bayes factors for non-nested models.	53
4.1	Example of induced Markov independence graph.	62
4.2	Symmetric Latin Square with unique diagonal.	64
4.3	Example of basis functions.	67
4.4	Combined graph from ten Facebook ego-nets.	71
4.5	Facebook data: GLM results.	72
4.6	Facebook data: Non-parametric results.	74
4.7	Facebook data: Residual analysis for non-parametric model.	75

4.8	Facebook data, ego-net 108: Non-parametric results.	77
4.9	Facebook data, ego-net 1,685: Non-parametric results.	78
4.10	Facebook data, ego-nets: Residual analysis for non-parametric model.	79
A.1	Example of induced Markov independence graph.	VIII

List of Tables

- 3.1 Karate club data: Model fitting results (triangle vs. random effect). 35
- 3.2 Karate club data: Model fitting results (GWESP vs. random effect). 39
- 3.3 Kapferer data: Model fitting results. 45
- 3.4 EP data: Model fitting results. 50
- 3.5 Simulation: Resulting log Bayes factors for nested models. 52

- 4.1 Facebook data: GLM results. 71
- 4.2 Facebook data: Non-parametric results. 73
- 4.3 Facebook data, ego-nets: Non-parametric results. 76

- A.1 Network Statistics with Markov assumption. VI
- A.2 Network Statistics without Markov assumption. VII

References

- Airoldi, E. M., Blei, D. M., Fienberg, S. E., and Xing, E. P. (2008). Generalized latin rectangles I: Construction and decomposition. *Journal of Machine Learning Research*, 9:1981–2014.
- Andersen, L. D. and Hilton, A. J. W. (1980). Generalized latin rectangles I: Construction and decomposition. *Discrete Mathematics*, 31:125–152.
- Bastian, M., Heymann, S., and Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*. Software version 0.8.1 beta.
- Bogomolny, A. (2016). Latin squares. Simple construction. From Interactive Mathematics Miscellany and Puzzles <http://www.cut-the-knot.org/arithmetic/latin2.shtml>. [Online; accessed: 04 March 2016].
- Bouranis, L., Friel, N., and Maire, F. (2015). Bayesian inference for misspecified exponential random graph models. *arXiv preprint arXiv:1510.00934*.
- Brandes, U. and Wagner, D. (2004). visone – analysis and visualization of social networks. In Jünger and Mutzel (2004), pages 321–340.
- Breslow, N. E. and Clayton, D. (1993). Approximate inference in generalized linear models. *Journal of the American Statistical Association*, 88(421):9–25.
- Butts, C. T. (2008). network: A package for managing relational data in r. *Journal of Statistical Software*, 24(2):1–36.
- Caimo, A. and Friel, N. (2011). Bayesian inference for exponential random graph models. *Social Networks*, 33(1):41–55.
- Caimo, A. and Friel, N. (2013). Bayesian model selection for exponential random graph models. *Social Networks*, 35(1):11–24.
- Caimo, A. and Friel, N. (2014). Bergm: Bayesian exponential random graphs in R. *Journal of Statistical Software*, 61(2):1–25.
- Chatterjee, S. and Diaconis, P. (2013). Estimating and understanding exponential random graph models. *The Annals of Statistics*, 41(5):2428–2461.
- Daudin, J. J., Picard, F., and Robin, S. (2008). A mixture model for random graphs.

- Statistics and Computing*, 18(2):173–183.
- Desmarais, B. A. and Cranmer, S. J. (2012a). Statistical inference for valued-edge networks: The generalized exponential random graph model. *PLoS ONE*, 7(1):1–12.
- Desmarais, B. A. and Cranmer, S. J. (2012b). Statistical mechanics of networks: Estimation and uncertainty. *Physica A: Statistical Mechanics and its Applications*, 391(4):1865–1876.
- Eilers, P. H. C. and Marx, B. D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121.
- Erdős, P. and Rényi, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen*, 6(290):290–297.
- Everitt, R. G. (2012). Bayesian parameter estimation for latent markov random fields and social networks. *Journal of Computational and Graphical Statistics*, 21(4):940–960.
- Fahrmeir, L. and Tutz, G. (2001). *Multivariate Statistical Modelling Based on Generalized Linear Models*. Springer New York, New York, NY, 2nd edition.
- Fellows, I. and Handcock, M. S. (2012). Exponential-family random network models. *arXiv preprint arXiv:1208.0121*.
- Fienberg, S. E. (2012). A Brief History of Statistical Models for Network Analysis and Open Challenges. *Journal of Computational and Graphical Statistics*, 21(4):825–839.
- Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, 80(1):27–38.
- Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842.
- Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185.
- Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F., and Hothorn, T. (2016). *mvtnorm: Multivariate Normal and t Distributions*. R package version 1.0-5.
- Geyer, C. J. and Thompson, E. A. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54(3):657–699.
- Gilbert, E. N. (1959). Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144.
- Gill, P. S. and Swartz, T. B. (2004). Bayesian analysis of directed graphs data with application to social networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53(2):249–260.
- Goldenberg, A., Zheng, A. X., Fienberg, S. E., and Airoldi, E. M. (2010). A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233.

-
- Goodreau, S. M. (2007). Advances in exponential random graph (p^*) models applied to a large social network. *Social Networks*, 29(2):231–248.
- Grün, B. and Leisch, F. (2008). Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software*, 28(1):1–35.
- Handcock, M. S. (2003). Assessing degeneracy in statistical models of social networks. *Technical report, Center for Statistics and Social Sciences, University of Washington*. <http://www.csss.washington.edu/Papers/wp39.pdf>.
- Handcock, M. S. and Gile, K. J. (2010). Modeling social networks from sampled data. *The Annals of Applied Statistics*, 4(1):5–25.
- Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., and Morris, M. (2008). statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*, 24(1):1–11.
- Hanneke, S., Fu, W., and Xing, E. P. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605.
- Hastie, T. and Tibshirani, R. (1990). *Generalized Additive Models*. Chapman & Hall, London.
- Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.
- Holland, P. W. and Leinhardt, S. (1981). An exponential family of probability distributions for directed graphs. *Journal of the American Statistical Association*, 76(373):33–50.
- Hummel, R. M., Hunter, D. R., and Handcock, M. S. (2012). Improving simulation-based algorithms for fitting ERGMs. *Journal of Computational and Graphical Statistics*, 21(4):920–939.
- Hunter, D. R. (2007). Curved exponential family models for social networks. *Social Networks*, 29(2):216–230.
- Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008a). Goodness of fit of social network models. *Journal of the American Statistical Association*, 103(481):248–258.
- Hunter, D. R. and Handcock, M. S. (2006). Inference in curved exponential family models for networks. *Journal of Computational and Graphical Statistics*, 15(3):565–583.
- Hunter, D. R., Handcock, M. S., Butts, C. T., Goodreau, S. M., and Morris, M. (2008b). ergm: A package to fit, simulate and diagnose exponential-family models for networks. *Journal of Statistical Software*, 24(3):1–29.
- Hunter, D. R., Krivitsky, P. N., and Schweinberger, M. (2012). Computational statistical methods for social network analysis. *Journal of Computational and Graphical Statistics*,

- 21(4):856–882.
- Jünger, M. and Mutzel, P. (2004). *Graph Drawing Software*. Springer Berlin Heidelberg.
- Kapferer, B. (1972). *Strategy and Transaction in an African Factory: African Workers and Indian Management in a Zambian Town*. Manchester University Press.
- Kass, R. E. and Raftery, A. E. (1995). Bayes Factors. *Journal of the American Statistical Association*, 90(430):773–795.
- Kauermann, G. (2005). Penalized spline smoothing in multivariable survival models with varying coefficients. *Computational Statistics & Data Analysis*, 49(1):169–186.
- Kauermann, G., Krivobokova, T., and Fahrmeir, L. (2009). Some asymptotic results on generalized penalized spline smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):487–503.
- Kolaczyk, E. D. (2009). *Statistical Analysis of Network Data: Methods and Models*. Springer, New York.
- Koskinen, J. and Daraganova, G. (2013). Dependence graphs and sufficient statistics. In Lusher et al. (2013), pages 77–90.
- Koskinen, J. and Snijders, T. A. B. (2013). Simulation, estimation, and goodness of fit. In Lusher et al. (2013), pages 141–185.
- Krivitsky, P. and Butts, C. T. (2015). Modeling valued networks with statnet. <http://statnet.csde.washington.edu/workshops/SUNBELT/previous/Valued/Valued.pdf>. [Online; accessed: 18 April 2016].
- Krivitsky, P. N. (2012). Exponential-family random graph models for valued networks. *Electronic Journal of Statistics*, 6:1100–1128.
- Krivitsky, P. N. and Handcock, M. S. (2008). Fitting position latent cluster models for social networks with latentnet. *Journal of Statistical Software*, 24(5).
- Krivitsky, P. N., Handcock, M. S., Raftery, A. E., and Hoff, P. D. (2009). Representing degree distributions, clustering, and homophily in social networks with latent cluster random effects models. *Social Networks*, 31(3):204–213.
- Krivitsky, P. N. and Kolaczyk, E. D. (2015). On the question of effective sample size in network modeling: An asymptotic inquiry. *Statistical Science*, 30(2):184–198.
- Krivobokova, T. and Kauermann, G. (2007). A note on penalized spline smoothing with correlated errors. *Journal of the American Statistical Association*, 102(480):1328–1337.
- Lazega, E. and Snijders, T. A. B. (2016). *Multilevel Network Analysis for the Social Sciences: Theory, Methods and Applications*. Springer International Publishing.
- Leisch, F. (2004). Flexmix: A general framework for finite mixture models and latent class

- regression in r. *Journal of Statistical Software*, 11(1):1–18.
- Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data/>.
- Lomi, A. and Stadtfeld, C. (2014). Social networks and social settings: Developing a coevolutionary view. *KZfSS Kölner Zeitschrift für Soziologie und Sozialpsychologie*, 66(1):395–415.
- López-Pintado, S. and Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486):718–734.
- Lubbers, M. J. and Snijders, T. A. B. (2007). A comparison of various approaches to the exponential random graph model: A reanalysis of 102 student networks in school classes. *Social Networks*, 29(4):489–507.
- Lusher, D., Koskinen, J., and Robins, G. (2013). *Exponential Random Graph Models for Social Networks*. Cambridge University Press, Cambridge.
- Maier, V., Furlinger, K., and Kauermann, G. (2016). A note on parallel sampling in exponential random graph models. Ms. LMU Munich.
- McAuley, J. J. and Leskovec, J. (2012). Learning to discover social circles in ego networks. In *Neural Information Processing Systems*.
- McCulloch, C. E., Searle, S. R., and Neuhaus, J. M. (2008). *Generalized, Linear, and Mixed Models*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2nd edition.
- Milgram, S. (1967). The small world problem. *Psychology Today*, 2(1):60–67.
- Morris, M., Handcock, M., and Hunter, D. (2008). Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software*, 24(1):1–24.
- Mosler, K. and Polyakova, Y. (2012). General notions of depth for functional data. *ArXiv e-prints*.
- Murray, I., Ghahramani, Z., and MacKay, D. (2006). MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, Arlington, Virginia. AUAI Press.
- Nowicki, K. and Snijders, T. A. B. (2001). Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087.
- O’Sullivan, F. (1986). A statistical perspective on ill-posed inverse problems. *Statistical Science*, 1(4):502–518.
- Pattison, P. and Snijders, T. A. B. (2013). Modelling social networks next steps. In Lusher et al. (2013), pages 287–301.

- Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). CODA: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11.
- R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Ramsay, J. O., Wickham, H., Graves, S., and Hooker, G. (2014). *fda: Functional Data Analysis*. R package version 2.4.4.
- Rinaldo, A., Fienberg, S. E., and Zhou, Y. (2009). On the geometry of discrete exponential families with application to exponential random graph models. *Electronic Journal of Statistics*, 3:446–484.
- Robbins, H. and Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407.
- Robins, G. and Lusher, D. (2013). Illustrations: Simulation, estimation, and goodness of fit. In Lusher et al. (2013), pages 167–185.
- Robins, G. L., Elliot, P., and Pattison, P. (2001). Network models for social selection processes. *Social Networks*, 23(1):1–30.
- Robins, G. L., Pattison, P., Kalish, Y., and Lusher, D. (2007a). An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):173–191.
- Robins, G. L., Snijders, T. A. B., Wang, P., Handcock, M. S., and Pattison, P. (2007b). Recent developments in exponential random graph (p^*) models for social networks. *Social Networks*, 29(2):192–215.
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.
- Ruppert, D., Wand, M. P., and Carroll, R. J. (2009). Semiparametric regression during 2003–2007. *Electronic Journal of Statistics*, 3:1193–1256.
- Salter-Townshend, M., White, A., Gollini, I., and Murphy, T. B. (2012). Review of statistical network analysis: models, algorithms, and software. *Statistical Analysis and Data Mining*, 5(4):243–264.
- Schall, R. (1991). Estimation in generalized linear models with random effects. *Biometrika*, 78(4):719–727.
- Schweinberger, M. (2011). Instability, Sensitivity, and Degeneracy of Discrete Exponential Families. *Journal of the American Statistical Association*, 106(496):1361–1370.
- Schweinberger, M., Handcock, M., and Luna, P. (2015). *hergm: Hierarchical Exponential-Family Random Graph Models with Local Dependence*. R package version 2.2-2.

- Schweinberger, M. and Handcock, M. S. (2015). Local dependence in random graph models: characterization, properties and statistical inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(3):647–676.
- Severini, T. (2000). *Likelihood Methods in Statistics*. Oxford science publications. Oxford University Press.
- Shalizi, C. R. and Rinaldo, A. (2013). Consistency under sampling of exponential random graph models. *The Annals of Statistics*, 41(2):508–535.
- Snijders, T. A. B. (1981). The degree variance: An index of graph heterogeneity. *Social Networks*, 3(3):163–174.
- Snijders, T. A. B. (2001). The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1):361–395.
- Snijders, T. A. B. (2002). Markov chain monte carlo estimation of exponential random graph models. *Journal of Social Structure*, 3(2):1–40.
- Snijders, T. A. B. (2016). The multiple flavours of multilevel issues for networks. In Lazega and Snijders (2016), pages 15–46.
- Snijders, T. A. B. and Koskinen, J. (2013). Longitudinal models. In Lusher et al. (2013), pages 130–140.
- Snijders, T. A. B., Koskinen, J., and Schweinberger, M. (2010a). Maximum likelihood estimation for social network dynamics. *The Annals of Applied Statistics*, 4(2):567–588.
- Snijders, T. A. B., Lomi, A., and Torló, V. J. (2013). A model for the multiplex dynamics of two-mode and one-mode networks, with an application to employment preference, friendship, and advice. *Social Networks*, 35(2):265–276.
- Snijders, T. A. B. and Nowicki, K. (1997). Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *Journal of Classification*, 14(1):75–100.
- Snijders, T. A. B., Pattison, P. E., Robins, G. L., and Handcock, M. S. (2006). New specifications for exponential random graph models. *Sociological Methodology*, 36(1):99–153.
- Snijders, T. A. B., van de Bunt, G. G., and Steglich, C. E. G. (2010b). Introduction to stochastic actor-based models for network dynamics. *Social Networks*, 32(1):44–60.
- Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639.
- Strauss, D. and Ikeda, M. (1990). Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 85(409):204–212.

- Sun, Y., Genton, M. G., and Nychka, D. W. (2012). Exact fast computation of band depth for large functional datasets: How quickly can one million curves be ranked? *Stat*, 1(1):68–74.
- The Inkscape Team (2015). Inkscape. <https://inkscape.org>. Software version 0.9.1.
- Thiemichen, S., Friel, N., Caimo, A., and Kauermann, G. (2016). Bayesian exponential random graph models with nodal random effects. *Social Networks*, 46:11–28.
- Thiemichen, S. and Kauermann, G. (2016). Stable exponential random graph models with non-parametric components for large dense networks. *arXiv preprint arXiv:1604.04732*.
- Turner, P. W., Kiel, M., and Schneider, M. (2013). Committee networks in the european parliament: Structure and impact on allocation of reports. Ms. LMU Munich.
- Turlach, B. A. and Weingessel, A. (2013). *quadprog: Functions to solve Quadratic Programming Problems*. R package version 1.5-5.
- van Duijn, M. A. J., Gile, K. J., and Handcock, M. S. (2009). A framework for the comparison of maximum pseudo likelihood and maximum likelihood estimation of exponential random graph models. *Social Networks*, 1(31):52–62.
- van Duijn, M. A. J., Snijders, T. A. B., and Zijlstra, B. J. H. (2004). p_2 : A random effects model with covariates for directed graphs. *Statistica Neerlandica*, 58(2):234–254.
- Varin, C., Reid, N., and Firth, D. (2011). An overview of composite likelihood methods. *Statistica Sinica*, 21(1):5–42.
- Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442.
- Whittaker, J. (2009). *Graphical Models in Applied Multivariate Statistics*. Wiley.
- Wood, S. (2006). *Generalized Additive Models*. Chapman & Hall, London.
- Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473.
- Zijlstra, B. J. H., Duijn, M. A. J., and Snijders, T. A. B. (2006). The multilevel p_2 model: A random effects model for the analysis of multiple social networks. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, 2(1):42–47.

Eidesstattliche Versicherung

(Siehe Promotionsordnung vom 12. Juli 2011, § 8 Abs. 2 Pkt. 5)

Hiermit erkläre ich an Eides statt, dass die Dissertation von mir selbstständig, ohne unerlaubte Beihilfe angefertigt ist.

München, den 24. Mai 2016

Stephanie Thiemichen

